

JUEGO DE ENSEÑANZA DE PROGRAMACIÓN PARA NIÑOS

**LUISA FERNANDA MARTINEZ FIGUEROA
HECTOR ORLANDO MOLINA SIERRA**

**FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
DEPARTAMENTO INGENIERÍA DE SISTEMAS**

BOGOTA, ABRIL 2017

JUEGO DE ENSEÑANZA DE PROGRAMACIÓN PARA NIÑOS

**LUISA FERNANDA MARTINEZ FIGUEROA
HECTOR ORLANDO MOLINA SIERRA**

TRABAJO DE GRADO

**DIRECTOR
CELIO GIL AROS**

**FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
DEPARTAMENTO INGENIERÍA DE SISTEMAS**

BOGOTÁ, ABRIL 2017

NOTA DE ACEPTACIÓN

Firma Jurado (Nombres)

Firma Jurado (Nombres)

Bogotá D.C., Abril de 2017

AGRADECIMIENTOS

Los autores del presente proyecto expresan sus agradecimientos:

A Dios en primer lugar, por iluminarnos en el logro de esta meta. Por ayudarnos en este proceso en el cumplimiento de cada aspecto de la carrera, por permitirnos esta grandiosa oportunidad de estudiar, de aprender de los docentes y compañeros.

A nuestras familias, por su apoyo incondicional. Gracias familia por nunca perder la fe en nosotros, confiar y creer en que podemos cumplir este sueño. La constancia, dedicación, persistencia y optimismo, son el propósito que nos inculcan ustedes día a día, lo cual permitieron que este gran paso, fuera posible, de igual forma nos enseñaron que luchar vale la pena.

A nuestro tutor de tesis Celio Gil Aros, por su enseñanza, paciencia, dedicación, confianza y tiempo. Gracias por ser un excelente guía en este proceso de culminación. Hemos aprendido mucho de usted.

A nuestros docentes por sus enseñanzas. Por su optimismo y constancia en el desarrollo del aprendizaje y por confiar en nosotros. Gracias porque ustedes nos ayudaron a cumplir esta meta. La exigencia, la comunicación, la enseñanza, la dedicación y la paciencia por parte de ustedes, permitieron que este sueño, uno de muchos, se cumpla.

A la Fundación Universitaria los Libertadores, por permitirnos cumplir este sueño de ser profesionales en su plantel educativo. Por enseñarnos la importancia de la ética de un profesional y la buena comunicación que como seres humanos debemos transmitir siempre. Gracias por enseñarnos a ser personas exitosas, con calidad humana y con objetivos claros en el desarrollo de este nuevo camino como ingenieros de sistemas.

DEDICATORIA

Este proyecto está dedicado a nuestros padres y hermanos, que siempre nos impulsaron a luchar, a estudiar y a persistir para culminar satisfactoriamente nuestra carrera como ingenieros de sistemas.

A los niños que disfrutaran del juego y que aprenderán una pequeña parte de un mundo inmenso que es la programación.

Tabla de contenido

Introducción	11
1. ASPECTOS DE LA INVESTIGACIÓN	12
1.1. Descripción del Problema	12
1.2. Formulación del problema.....	13
1.2.1. Población de interés.....	14
1.2.2. Pregunta de investigación.....	14
1.3. Justificación del proyecto de investigación.....	14
1.3.1. Comportamiento población de interés.	15
1.4. Delimitaciones.....	16
1.4.1. Cronograma.....	16
1.4.2. Conceptual.....	17
1.4.3. Recursos.....	18
1.5. Objetivos	19
1.5.1. General.	19
1.5.2. Específicos.	19
2. MARCO TEÓRICO.....	21
2.1. Introducción marco teórico.....	21
2.2. Antecedentes	22
2.2.1. Alice	22
2.2.2. Snake Wrangling for Kids.....	22
2.2.3. Kids with Ruby.	23
2.2.4. Kodu.	23
2.2.5. Code Monkey.....	23
2.2.6. Tarjetas Arduino, Raspberry PI, Lego Mindstorms NXT.	23
2.2.7. Legales.....	24
2.2.8. Investigativos.....	25
2.3. Bases teóricas.....	26
2.3.1. Tipo de Teoría de Aprendizaje:	26
2.3.2. Como aprenden los niños.	27

2.3.3.	Teorías genéricas basadas en ingeniería.	28
2.3.4.	Lenguajes de programación e ingeniería de software.....	29
2.3.5.	Lenguajes de animación y motores de videojuegos.....	30
2.3.6.	Construcción del marco conceptual.....	30
2.4.	Metas a alcanzar	31
2.5.	Principios	31
2.5.1.	Enfoque.....	31
2.5.2.	Productos del Modelo	32
2.5.3.	Controles.....	32
2.5.4.	Glosario	32
3.	DISEÑO METODOLOGICO	37
3.1	Tipo de Investigación	37
3.2.	Metodología Seleccionada	37
3.2.1.	Análisis de Necesidades	38
3.3.	Requerimientos del nuevo sistema	39
3.3.1.	Requerimientos funcionales	39
3.3.2.	Requerimientos no funcionales.....	39
3.4.	Diseño del nuevo sistema	40
3.4.1.	Diseño mapa de navegación.....	41
3.4.2.	Diseño orientado a objetos.....	42
3.4.3.	Diseño de casos de uso	42
3.4.4.	Diagrama de clases.....	49
3.4.5.	Diagrama de secuencia	50
3.4.6.	Diagrama de Actividades	55
3.4.7.	Diagrama de Paquetes.....	56
3.4.8.	Diagrama de Componentes.....	57
3.4.9.	Diagrama de Despliegue.....	58
3.5.	Diseño de interface de usuario	59
4.	ANÁLISIS DE RESULTADOS Y CONCLUSIONES	66
4.1.	Detalle de resultado	66
4.1.1.	Pruebas de Función	72
4.1.2.	Pruebas de Caja Blanca.....	77

4.1.3.	Pruebas Modulares	77
4.1.4.	Pruebas del Sistema.....	78
4.1.5.	Prueba de Interfaz.....	78
4.1.6.	Prueba de Calidad.....	78
4.1.7.	Tipos de pruebas realizadas.....	79
4.2.	Análisis de Resultados	79
4.3.	Conclusiones	80
4.4.	Recomendaciones.....	80
5.	Bibliografía.....	81
	ANEXOS	83

Lista de Tablas

<i>Tabla 1: Percepción Juego</i>	15
<i>Tabla 2: Estadística de uso del Juego</i>	15
<i>Tabla 3: Percepción Juego 2</i>	16
<i>Tabla 4: Estadística de percepción</i>	16
<i>Tabla 5: Recursos de Hardware</i>	18
<i>Tabla 6: Recursos de Software</i>	18
<i>Tabla 7: Recurso Humano</i>	19
<i>Tabla 8 Tipos de pruebas realizadas</i>	79

Lista de Ilustraciones

Ilustración 1: Estadística Uso del Juego	15
Ilustración 2: Mayor Creatividad	16
Ilustración 3 – Metodología	38
Ilustración 4 Mapa de Navegación	42
Ilustración 5 Caso de uso general.....	43
Ilustración 6 Caso de uso Inicio de sesión niños de 8 a 12 años	45
Ilustración 7 Caso de uso Nivel I 5 a 8 años	46
Ilustración 8 Caso de uso Nivel II 5 a 8 años	47
Ilustración 9 Caso de Uso Niveles 8 a 12 años	48
Ilustración 10 Diagrama de Clases	49
Ilustración 11 Diagrama de Secuencia	50
Ilustración 12 Diagrama Juego 5 a 8 años Nivel I y Nivel II	51
Ilustración 13 Diagrama Juego Nivel 1 8 a 12 años	52
Ilustración 14 Diagrama Juego Nivel 2 8 a 12 años.....	53
Ilustración 15 Diagrama Nivel 3 8 a 12 años	54
Ilustración 16 Diagrama de Actividades.....	55
Ilustración 17 Diagrama de Paquetes.....	56
Ilustración 18 Diagrama de Componentes.....	57
Ilustración 19 Diagrama de Despliegue.....	58

Introducción

El siguiente proyecto está enfocado a la realización de un videojuego, en el cual le enseñe al niño según su edad (5 años a 12 años), conceptos básicos y principales en el mundo de la programación, detectar objetos e inicializarlos, aprender sobre ciclos (for y while), además uno de los aspectos importantes de la programación es la ejecución de procesos y secuencias mediante una tarea o un objetivo. El juego, tendrá personajes llamativos y reconocidos para el niño. Mediante los sonidos se busca que el juego sea interesante y le permita al niño tener una experiencia cercana con los conceptos básicos de la programación. En la actualidad los videojuegos forman un papel importante en la sociedad tecnológica, haciendo que todas las personas ya sean niños y adultos se sumerjan en el mundo de los videojuegos. Si bien, lo anterior se afirma, gracias al estudio realizado por la Universidad del País Vasco 2011, dicho estudio elaborado por el profesor José Inazio Imaz indica en su libro ***"Influencia de la pantalla en la enseñanza: el caso de los videojuegos"*** que con la ayuda de los videojuegos se evidencia una mejora en lo que él llama ***"alfabetización digital"***, es decir, que con el uso de dichos juegos, los niños aprenden a desenvolverse en el ámbito social. Imaz afirma que los videojuegos no son dañinos como la mayoría piensa, de lo contrario, los juegos y más aún si son educativos ***"«Ayudan a desarrollar las competencias sociales de la sociedad que nos ha tocado vivir, ayudan a lograr la alfabetización digital»"***¹ indica Imaz. El estudio se tomó en cuenta con un total de 433 alumnos entre 11 a 18 años, evidenciando incluso mayor consumo de videojuegos en niños menores de 18 años. Por lo tanto los videojuegos permiten que los niños aprendan de una forma didáctica, dinámica y llena de emociones multi-sensoriales en algunos casos. Si bien se conoce que la informática es un campo inmenso de muchos conocimientos, es interesante detenerse en una de sus raíces y analizarla y aprender de esta misma. Una de las ramas de la informática es la programación y con el dinamismo de un videojuego se especifica a continuación, PIPOO 1.0.

¹ <http://gara.naiz.eus/paperezkoa/20110129/245501/es/Los-videojuegos-ayudan-desarrollar-competencias-sociales>

1. ASPECTOS DE LA INVESTIGACIÓN

1.1. Descripción del Problema

Ante la complejidad de los diversos problemas cotidianos que se generan día a día, es necesario que desde la infancia se construya unas bases de desarrollo del pensamiento lógico para que dichos problemas puedan ser comprendidos, asimilados y adaptados para encontrar la más óptima solución con las herramientas y recursos al alcance, tal como lo expone Jean William Fritz Piaget psicólogo suizo, en su estudio de la infancia y por su teoría constructivista del desarrollo de la inteligencia en donde se evidencia, con el incremento del uso de la tecnología y el auge de las aplicaciones móviles para que los niños a través de la lógica generada en el uso de los juegos didácticos virtuales, apoyados en las estructuras básicas de programación; en consecuencia se puede incrementar en oportunidades de aprendizaje y así mismo la resolución de problemas complejos a través del pensamiento lógico. Los problemas surgen de cualquier evento o situación, que realmente pongan en apuros al ser humano. Dichos problemas son solucionados por parte de los adultos en forma eficiente y logran en muchas ocasiones brindar una solución acertada y confiable según la situación. Sin embargo los niños hoy en día, en su mayoría, no desarrollan el nivel de resolución de problemas, e independientemente a esto, el desarrollo cognitivo de los niños es relevante en su aprendizaje. De igual forma el análisis, el conocimiento sobre la comunicación máquina – hombre e incluso el pensar diferente son ítems importantes en el crecimiento diario de un niño. Actualmente, los niños disfrutan de cerca las novedades de la tecnología y se sumergen cada vez más a aplicaciones y juegos por diversión. Sin embargo es desconocido para ellos como realizar dichas aplicaciones o cómo funcionan estos juegos. Por tal motivo los niños deberían de igual forma aprender sobre la programación de aplicaciones. Ya que según en la página del tiempo escrito por TECNOSFERA uno de los beneficios que ofrece la

programación es “permite aprender a interactuar con la tecnología, crearla y expresarla a través de diferentes mecanismos.” Mediante la resolución de problemas, el cual ayudaría al niño a interpretar mejor una situación, analizar el problema y brindar una solución. De igual forma el desarrollo cognitivo del niño incrementa y su nivel de análisis se formaría desde temprana edad. Mediante los términos básicos de programación como: condiciones, ciclos, funciones y objetos, los niños podrán a futuras generar páginas y pequeñas aplicaciones que les ayudará en su aprendizaje incluso en otras áreas.

1.2. Formulación del problema

Para contrarrestar la baja participación en las ingenierías y aún más en las de desarrollo tecnológico es muy importante encontrar un punto de equilibrio para dar un resultado a mediano y largo plazo como lo es la población infantil en etapa de reconocimiento de su mundo y con las habilidades para manipular las herramientas de tecnología. Para la población de hace dos décadas el acceso y facilidad a este tipo de herramientas era muy escaso sin embargo actualmente los infantes poseen una facilidad innata para la manipulación y aprendizaje de los dispositivos tales como los teléfonos inteligentes y los computadores personales; por tal motivo aprovechando dichas condiciones enfocando una aplicación de escritorio desarrollada en .NET se podrá facilitar la tarea de enseñar a programar de manera simulada, sencilla y didáctica a niños entre los 5 a 12 años de edad, a través de resolución de problemas (ABRP) de juegos mentales y objetivos los cuales inconscientemente sembraran el interés por la lógica matemática y programación a través de ciclos para lograr una meta. Es muy útil para los niños, ya que con el trabajo y desarrollo continuo de cada ejercicio propuesto se mantiene activo mentalmente al usuario.

1.2.1. Población de interés.

- ❖ Niños entre 5 a 8 años de edad
- ❖ Niños de 8 a 12 años de edad
- ❖ Posibles interesados para cualquier edad.

1.2.2. Pregunta de investigación.

¿De qué manera a través de un juego didáctico se puede desarrollar la creatividad en los niños y que en un futuro puedan utilizar esto como herramientas de programación?

Complementando los anteriores argumentos se ha comprobado que la programación en temprana edad aumenta la conexión neuronal al igual que el aprender a tocar un instrumento de música o aprender un nuevo idioma; con un interés propuesto hacia la solución de problemas a través de una aplicación de escritorio un niño obtendrá una mayor concentración, reafirmar su carácter en la toma de decisiones y actitud ante posibles obstáculos lo que para su vida adulta ayudará para la formación de una personalidad definida.

Para la sociedad en general el crecimiento depende de las áreas que más desarrollo presenten y la ingeniería de sistemas es una carrera en continuo cambio y evolución; al inducir un interés a las nuevas generaciones se aumentará el número de nuevos estudiantes y profesionales futuros a su vez la creación de nuevas formas de pensamiento e innovación. Adicionalmente si se aplica este tipo de ejercicios a personas de mayor edad se evitará el padecimiento de enfermedades como la pérdida de memoria como el síndrome de Alzheimer.

1.3. Justificación del proyecto de investigación

Mediante una aplicación web PEQUEÑOS INGENIOS PROGRAMACIÓN ORIENTADA A OBJETOS PIPOO 1.0, se podrá evaluar el desarrollo y destreza de un niño entre los 5 a 12 años de edad a través de la resolución de problemas de lógica. Al desarrollar un pensamiento lógico psicológicamente se genera mayor confianza en la toma de decisiones lo que implica un gran apoyo para la vida social y personal de una persona.

1.3.1. Comportamiento población de interés.

La población a la cual está dirigida la investigación está en el rango de 5 a 12 años. En el cual se tiene en cuenta el nivel de confianza (Z), la probabilidad de éxito (P) y la probabilidad de error (Q). Se especifica a continuación mediante la fórmula expuesta $n = \frac{Z^2 \cdot P \cdot Q}{D^2}$. La margen de confianza y la probabilidad de error son altos, por lo tanto, no se determina una cantidad definida para la muestra. Se tiene de igual forma una validación del juego, probada por 20 niños, de esta muestra se pudo determinar la media, moda, varianza, etc. Datos estadísticos que ayudaron a concluir que el 30% de los niños usaron el juego por curiosidad y el 25% por diversión.

Toma de la muestra

Z = Nivel de confianza
P = Probabilidad de éxito
Q = Probabilidad de error

$$n = \frac{Z^2 \cdot P \cdot Q}{D^2}$$

$$n = \frac{1,96^2 \cdot 0,005 \cdot 0,95}{0,03^2}$$

$$n = 203 = 20,3 \text{ (10\%)}$$

Uso del juego	Datos
Interés	4
Diversión	5
Curiosidad	6
Obligación	0
No gusta	0
No sabe	5

Tabla 1: Percepción Juego

MUESTRA	20
MEDIA	3.33333333
MODA	5
MEDIANA	3
VARIANZA	7.06666667
DESVIACION ESTÁNDAR	2.65832027

Tabla 2: Estadística de uso del Juego

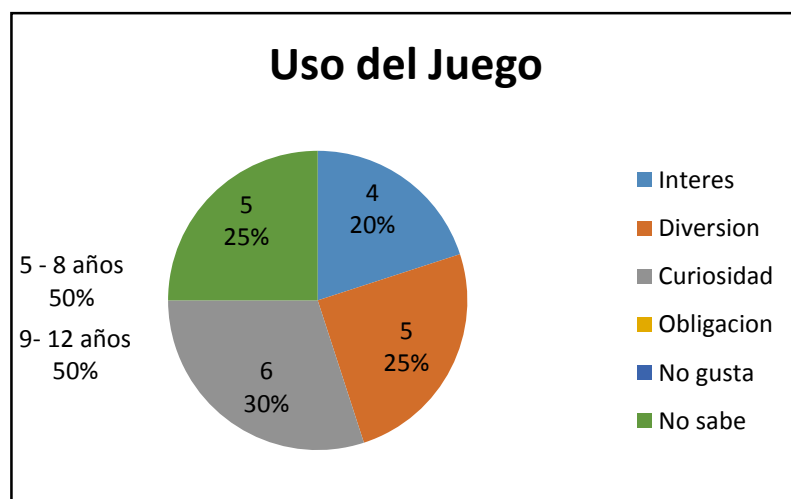


Ilustración 1: Estadística Uso del Juego

Una vez usado el juego, se les solicito crear un dibujo libre. 15 niños demostraron mayor creatividad a la hora de exponer sus dibujos. Solo 5 alumnos correspondientes a un 25 % de la totalidad no demostraron creatividad.

Mayor creatividad	Datos
Si	15
No	5

Tabla 3: Percepción Juego 2

MUESTRA	20
MEDIA	10
MODA	No hay moda
MEDIANA	10
VARIANZA	50
DESVIACION ESTÁNDAR	7.071067812

Tabla 4: Estadística de percepción

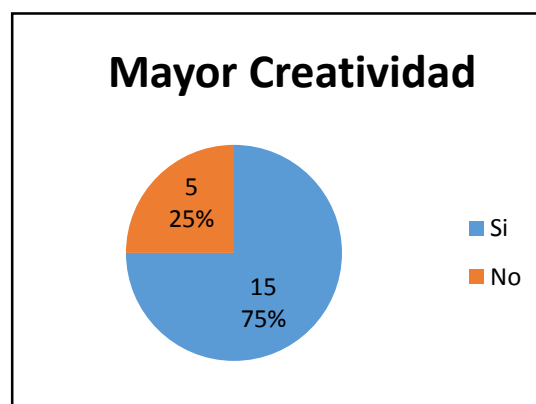


Ilustración 2: Mayor Creatividad

Interés del estudio:

Observar el comportamiento de la población sobre el uso y el aumento de la creatividad.

1.4. Delimitaciones

- El juego contará con 6 niveles, segmentados en: 3 niveles para niños de 5 a 8 años y 3 niveles para niños de 8 a 12 años
- El juego PIPOO 1.0 tendrá niveles de aprendizaje de: laberintos, desplazamiento de figuras (runner) y encontrar objetos.

1.4.1. Cronograma.

El proyecto tendrá un desarrollo cronológico de seis (6) meses calendario. (Se prorrogará según el nivel deseado a (9) meses si es necesario para mejorar la aplicación del proyecto). Se anexa cronograma de actividades.

1.4.2. Conceptual.

Siguiendo la metodología de ²Álvaro Galvis Panqueva, se contempla que el juego PIPOO 1.0 criterios para evaluar e incentivar la creatividad en los niños y su potencial en la adopción de un pensamiento lógico analítico. Así, mismo al verificar como guía principal en el artículo se resalta la importancia en el entorno en el cual sea presentado el juego, es decir: los colores, el sonido, la creatividad que se usó para llegar a dicho fin del juego. En el artículo Ambientes Educativos, en la página No. 27, se indica la importancia de añadir lúdica a un juego de enseñanza *“Es a partir de encontrar evidencias de que hemos sido lúdicos, creativos y colaborativos cómo podemos rescatar la autoestima de quienes, como en el caso de niños de zonas marginales y en situación de riesgo, suelen tenerla muy lastimada”*. De igual forma el nivel de dificultad es importante al momento de realizar un juego, ya que cada nivel debe presentar mayor dificultad que el anterior, en este sentido se expresa *“así como de aprender a resolver problemas de cierta complejidad, donde los múltiples puntos de vista son necesarios.”* de este modo, se evidencia la importancia de enseñar a su vez al niño a solucionar inconvenientes que se le puedan presentar. Por lo tanto es importante tener como base dicho artículo para el desarrollo del juego PIPOO 1.0, para optimizar funcionalidades y procesos internos en el juego.

El ciclo de vida del software impacta al proyecto de desarrollo por lo cual se deben tomar en cuenta los siguientes aspectos:

- ❖ Ingeniería de requerimientos
- ❖ Análisis de requerimientos
- ❖ Diseño y modelaje del proyecto
- ❖ Construcción del software
- ❖ Pruebas funcionales.
- ❖ Documentación Manual de usuario.

² http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-106223_archivo.pdf

1.4.3. Recursos.

Recursos de hardware: Se describe a continuación los recursos tangibles para la realización de proyecto durante su desarrollo y planeación

HARDWARE			
ITEM	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Portatil Samsung 300E; 2,40 GHZ; CORE I 5;2 GB RAM; WINDOWS 7	1 UN	\$ 1'500.000	\$ 1'500.000
Portatil Lenovo G410; 2,40 GHZ; CORE I 7;8 GB RAM; WINDOWS 10	1 UN	\$ 1'500.000	\$ 1'500.000
TOTAL			\$ 3'000.000

Tabla 5: Recursos de Hardware

Recursos de software: Para la fabricación del proyecto de software, se toma en cuenta los recursos no tangibles que son esenciales el proceso de programación.

SOFTWARE			
ITEM	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
MODELAMIENTO (UML) LIBRE	1 UN	\$ 0	\$ 0
BASE DE DATOS SQL SERVER 2008, SERVER FREE	1 UN	\$ 0	\$ 0
ASP .NET C#, USER OUTLOOK - UNITY	1 UN	\$ 0	\$ 0
TOTAL			\$ 0

Tabla 6: Recursos de Software

Recurso humano: Se contempla en el proceso de planeación creación y culminación; al recurso humano como parte principal de la investigación para cada método, proceso y escala del desarrollo.

Descripcion	Horas x Semana	Horas Trabajadas
Analisis del problema	20	12 (semanal)
Levantamiento de requerimientos Luisa y Orlando	20	8 (semanal)
Diseño arquitectonico	20	40 (x dos semanas)
Configuracion e instalacion de herramientas	20	20 (semanal)
Desarrollo de la aplicación	20	320 (x cuatro meses)
Ajustes de la aplicación	20	40 (mensual)
Pruebas de la aplicación	20	20 (semanal)
Ajustes de la aplicación	20	40 (mensual)
Analisis Final	20	40 (mensual)
Ingeniero Hora	80000 x hora	540 Horas
	\$43,200,000	

Tabla 7: Recurso Humano

1.5. Objetivos

1.5.1. General.

Diseñar y desarrollar un juego que permita incentivar el aprendizaje de la programación básica en niños entre 5 a 12 años.

1.5.2. Específicos.

- Diseñar un estándar de desarrollo sencillo y útil para el niño según las edades.
- Clasificar los escenarios de dificultad en el juego, para aumentar el mecanismo de solución de problemas.
- Realizar juegos, retos y resolución de problemas en un ambiente divertido y didáctico en pos de aumentar el interés por la programación y ciclos

lógicos.

- Identificar por parte del niño las propiedades básicas de la programación como: Ciclos, condiciones, funciones y objetos.
- Aplicar la metodología SEI de Álvaro Galvis y desarrollar criterios importantes en el juego con base a la metodología estudiada.

2. MARCO TEÓRICO

2.1. Introducción marco teórico

En la actualidad, los juegos de los niños cada vez van siendo más tecnológicos, según el estudio realizado por la universidad de Wisconsin, en Estados Unidos del 2013. Los juguetes incorporan sonidos llamativos o imágenes y colores atractivos al niño. Sin embargo la tecnología también incorpora pasos gigantes en el desarrollo cognitivo de los niños. Ya que los juegos hoy en día no solo son como fragmento de diversión sino de aprendizaje también. La tecnología ha incrementado la perspectiva de aprendizaje sobre la informática, no es aprender sobre computadoras sino la lógica que hay en ellos, sin embargo aunque suene complejo, hay que determinar qué es importante para el niño desarrollar su nivel de aprendizaje cognitivo, de lógica y razonamiento, por medio de la programación y la creación de algoritmos, sencillos pero muy provechosos. Es importante entender que hay muchos tipos de juegos que influyen en el desarrollo de razonamiento y lógica, los juegos deben tener un nivel de dificultad, como lo indican en el artículo “Ambientes Educativos para la Era Informática” de Álvaro H Galvis³, hace referencia a las actividades lúdicas, creativas y de solución de problemas, en el cual indican que cuando se realiza actividades en el juego, se puede estudiar incluso la personalidad de la persona, por lo que realiza dichas actividades con una destreza natural. En el mismo artículo resalta una palabra no menos importante que las anteriores nombradas, esta es la “*Ludopedagogía*”, en el cual mediante un ejemplo indica factores importantes a tener en cuenta en el desarrollo de un juego, etapas que deben seguir cierta sintonía con toda la base del juego y así lograr el éxito del mismo con el usuario final. Los elementos a tener en cuenta son: **Inspirador** (Contextualizar al usuario desde un principio, si es necesario contar mediante imágenes alguna historia sobre la representación del juego), **Juego – Actividad** (Se

³ www.colombiaaprende.edu.co/html/mediateca/1607/articles-88541_archivo.pdf

generan los niveles o etapas por el cual el usuario pasará por cada uno, con un grado de dificultad cada vez mayor y la obtención de recompensas), **Reconocimiento** (Depende del juego se puede determinar ciertas preguntas o suposiciones sobre el tema del juego y luego las respuestas nos llevarán a una conclusión), **Recreación** (Ya se tiene una experiencia y hechos sobre el juego, por lo que es positivo crear grupos de intereses entre los niños, para que entre ellos compartan sus experiencias). Es importante que en el desarrollo de un juego para niños se tenga presente muchos factores como los colores, texturas y sonidos, para que estos elementos ayuden al estímulo de los sentidos y la ejecución de actividades dentro del juego.

2.2. Antecedentes

Hay una serie de aplicaciones destinadas al aprendizaje de la programación, a continuación se citarán las más relevantes, entre las cuales tenemos:

2.2.1. Alice⁴.

Es una aplicación que enseña al niño a programar por medio del uso de bloques de 3D, no solo es programar sino crear historias también.

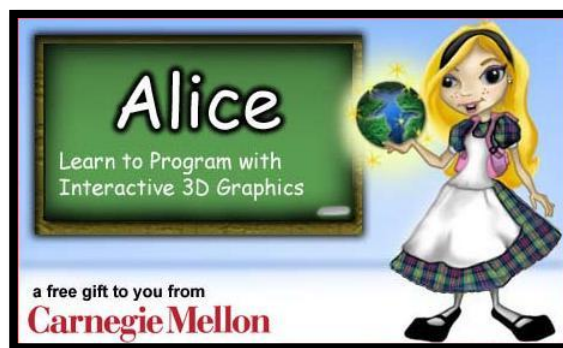


Imagen 1 Fuente: url:
<http://www.alice.org/index.php?page=d>

2.2.2. Snake Wrangling for Kids⁵.

Es un libro electrónico que le enseña a los niños mayores de 8 años a programar. Es un lenguaje de programación que usa Python. Es una herramienta sencilla y versátil para



Imagen 2 Fuente: url:www.briggs.net.nz/snake-wrangling-for-kids.html

⁴ url: <http://www.alice.org/index.php?page=d>

⁵ url:www.briggs.net.nz/snake-wrangling-for-kids.html

el uso de los niños.

2.2.3. Kids with Ruby6.

Creado por Yukihiro Matsumoto, es uno de los programas más interactivos, originales y divertidos, en el cual se aprende a programar de forma didáctica y genera interés en jóvenes programadores.

2.2.4. Kodu7.

Una aplicación diseñada para el aprendizaje de programación y la creación de juegos desde PC y Xbox. Muy útil y de gusto de los niños.

2.2.5. Code Monkey8.

Es un juego didáctico en el cual presenta a Monkey, el personaje tendrá que avanzar para recoger bananas, para hacerlo avanzar es necesario escribir unas líneas de código sencillas, que permiten al usuario familiarizarse con los lenguajes de programación, su estructura y la lógica que estos programas usan.

2.2.6. Tarjetas Arduino, Raspberry PI, Lego Mindstorms NXT.

Estos dispositivos demuestran que no solo las líneas de código se puede aprender sencillamente, sino que la construcción de robots ayudan al desarrollo cognitivo de

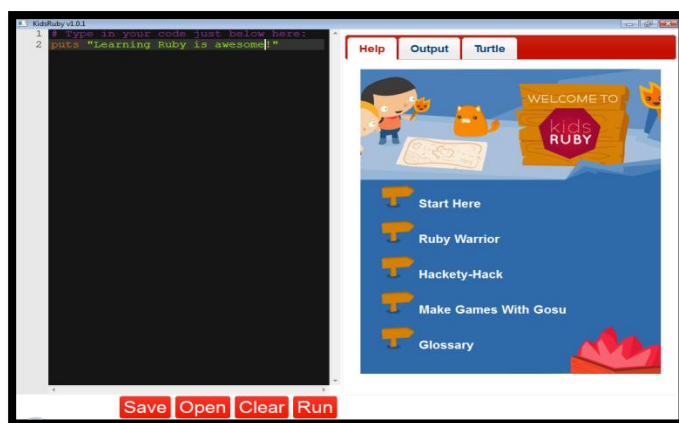


Imagen 3 Fuente: url:kidsruby.com/



Imagen 4 Fuente: [url: https://www.microsoft.com/en-us/download/details.aspx?id=10056](https://www.microsoft.com/en-us/download/details.aspx?id=10056)



Imagen 5 Fuente: [url: https://www.playcodemonkey.com/](https://www.playcodemonkey.com/)

⁶ url:kidsruby.com/

⁷ [url: https://www.microsoft.com/en-us/download/details.aspx?id=10056](https://www.microsoft.com/en-us/download/details.aspx?id=10056)

⁸ [url: https://www.playcodemonkey.com/](https://www.playcodemonkey.com/)

los niños. Los lenguajes de programación que usan dichos dispositivos son de igual forma sencillos y versátiles.

Si bien cabe resaltar la importancia de estas aplicaciones para que los niños conozcan cómo se puede solucionar un problema desde muchas perspectivas y es que la programación no es solo programar y ya, como lo especifica Pablo Espeso en su blog de “Como iniciar a un niño en la programación desde cero” indica: *“La programación es una herramienta que puede fomentar la creatividad a unos niveles extraordinarios, siempre que se sepa cómo hacerlo. No se trata sólo de resolver problemas, si no de utilizar los recursos y las herramientas disponibles para realizar las creaciones que tengamos en la mente. No se trata sólo de saber cómo se programa un Pong, si no de que una vez lo hayas programado sepas ir un paso más allá. Sepas añadir marcadores a cada jugador, tengas la intención de añadir un ranking con los mejores jugadores, o de dar la posibilidad de elegir el color del juego”*.⁹ Es realmente importante conocer hasta qué límites o hasta donde se puede llegar con la programación y que esta misma ayude a los niños a crecer con un nivel de conocimiento más amplio. Si bien límites no habría en la programación, todo depende de la imaginación que conlleve las creaciones¹⁰.

2.2.7. Legales

A continuación se dará a conocer los aspectos legales que rigen el desarrollo de productos de software:

- **Ley 23 de 1982**¹¹. Los autores de obras literarias, científicas y artísticas gozarán de protección para sus obras en la forma prescrita por la presente ley y, en cuanto fuere compatible con ella, por el derecho común. También protege esta ley a los intérpretes o ejecutantes, a los productores de fonogramas y a los organismos de radiodifusión, en sus derechos conexos a los del autor.

⁹ Pablo Espeso, España Valladolid, url <http://www.xataka.com/otros/como-iniciar-a-un-nino-en-la-programacion-desde-cero>

¹⁰ url: <http://www.xataka.com/otros/como-iniciar-a-un-nino-en-la-programacion-desde-cero>

¹¹ www.derechodeautor.gov.co/HTM/legal/jurisprudencia/Corte%20Constitucion

- **Leyes Colombianas. Ley 44 De 1993.** Especifica penas entre dos y cinco años de cárcel, así como el pago de indemnizaciones por daños y perjuicios, a quienes comentan el delito de piratería de software. Se considera delito el uso o reproducción de un programa de computador de manera diferente a como está estipulado en la licencia. Los programas que no tengan licencia son ilegales. Es necesaria una licencia por cada copia instalada.

2.2.8. Investigativos

En la elaboración del estado del arte tendremos los siguientes artículos:

El artículo del Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias de matemáticas en primaria. (Carlos Andrés Palma Suárez / Román Eduardo Sarmiento Porras) fue expuesto en el 2015 en la revista mexicana de investigación educativa. Este artículo indaga sobre las posibilidades de mejoramiento que los algoritmos, pueden generar en los niños que cursen 5° de primaria. Se plantea generar un nuevo modelo educativo, en el cual mediante la programación los niños mejoren sus habilidades y reducir los estados de deterioro en la educación sobre todo en el área de las matemáticas. Sin embargo para crear este nuevo módulo, es necesario realizar un plan piloto e introducir poco a poco el mecanismo de pasos y la lógica matemática. Como lo indica en el artículo “Uno de los aspectos que abarca la enseñanza tecnológica es la comprensión de los procesos lógicos como fases secuenciales de operaciones que transforman recursos, el fin es cumplir con ciertos objetivos y lograr un resultado esperado”¹², esto último referencia no solo la importancia del aprendizaje en la comprensión de cada paso y proceso sino en la resolución de problemas. Si bien sabemos que las matemáticas son de gran utilidad en hallazgos y procesamiento de problemas, también lo es la programación. Incluso en el artículo se menciona que “Abelson, Sussman y Sussman (1996) abordan la importancia de la enseñanza de la programación de computadores, a tal punto de presentarla como un complemento de la matemática, dado que mientras “la matemática provee un

¹² <https://drive.google.com/drive/my-drive>

marco de trabajo para tratar en forma precisa con nociones de tipo -qué-, la computación provee un marco de trabajo para tratar en forma precisa con nociones de tipo -cómo”¹³ . De igual forma si se distinguen todas las aplicaciones creadas para fines didácticos para el aprendizaje de los niños, todas cumplen con un requisito en particular, *“se enseñó la organización de un conjunto de órdenes secuenciales para la ejecución de tareas particulares.”* (Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias de matemáticas en primaria, 2015).

Uno de los factores importantes que resalta el artículo es la edad. Por lo cual se realiza la pregunta ¿hay alguna edad en específico para empezar a aprender a programar?, según Michael Kölling (citado en Utting et al., 2010), *“no se presenta un rango especial de edades para la enseñanza de la programación en general”*, incluso asegura que *“entre más temprano se enseñe, mucho mejor.”*

Es un artículo interesante desde todas las perspectivas y recomendable para su lectura.

2.3. Bases teóricas

Teorías de Aprendizaje:¹⁴ “son aquellas que realizan la descripción de un proceso que permite que una persona o un animal aprendan algo. Estas teorías pretenden entender, anticipar y regular la conducta a través del diseño de estrategias que faciliten el acceso al conocimiento.”

2.3.1. Tipo de Teoría de Aprendizaje:

La Teoría del Aprendizaje de Jean Piaget - Psicología y Mente

Jean Piaget es uno de los más conocidos psicólogos del enfoque constructivista, una corriente que bebe directamente de las teorías del aprendizaje de autores

¹³ <https://drive.google.com/drive/my-drive> , Pagina 609, Revista Mexicana de Investigacion Educativa

¹⁴ <definicion.de/teoria-del-aprendizaje/>

como Lev Vygotsky o David Ausubel. El enfoque constructivista, en su vertiente de corriente pedagógica, es una manera determinada de entender y explicar las formas en las que aprendemos. Los psicólogos que parten de este enfoque ponen énfasis en la figura del aprendiz como el agente que en última instancia es el motor de su propio aprendizaje. ¿Por qué se dice que Piaget es constructivista? En términos generales, porque este autor entiende el aprendizaje como una reorganización de las estructuras cognitivas existentes en cada momento. Es decir: para él, los cambios en nuestro conocimiento, esos saltos cualitativos que nos llevan a interiorizar nuevos conocimientos a partir de nuestra experiencia, se explican por una recombinación que actúa sobre los esquemas mentales que tenemos a mano tal como nos muestra la Teoría del Aprendizaje de Piaget.

El concepto de esquema es el término utilizado por Piaget a la hora de referirse al tipo de organización cognitiva existente entre categorías en un momento determinado. Es algo así como la manera en la que unas ideas son ordenadas y puestas en relación con otras.

2.3.2. Como aprenden los niños¹⁵.

Los niños aprenden desde el vientre de sus madres a través de estímulos, después de nacer todo lo que aprenden es por los sentidos como el tacto, olfato etc. A medida que crecen ellos relacionan los objetos y los colores, aprenden por repetición, asociación. Los padres tienen un papel muy importante en el proceso del aprendizaje del niño ya que el niño aprende lo que los padres hacen, como caminar, moverse, hablar, comer todo si los padres tienen buenos hábitos los niños los tendrá, cabe aclarar que el niño no solo aprende de sus padres i no de todo aquello que lo rodea, en nuestro nuevo mundo los niños aprende de los medios de comunicación televisión, Internet, tecnología y es por esto que los niños de hoy en día se les facilita el manejo de las nuevas herramientas tecnológicas pues porque desde pequeños están en contacto con ellas.

¹⁵ : <http://aprendizajeautonomo2.bligoo.com.co/conclusiones-como-aprenden-los-ninos#.WH6AONizV8w>

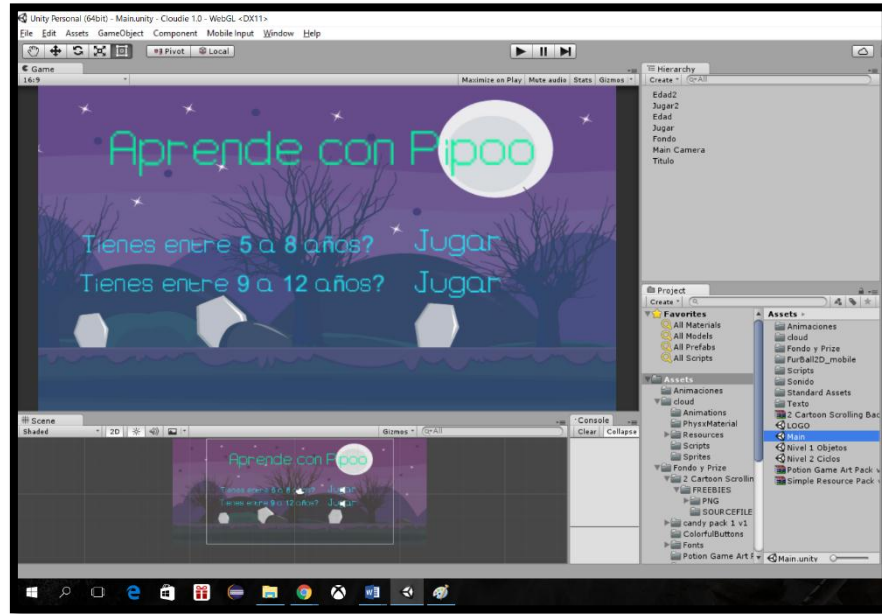
2.3.3. Teorías genéricas basadas en ingeniería.

La ingeniería de sistemas abarca muchas funciones, términos y ramas en el cual se usan en el día a día

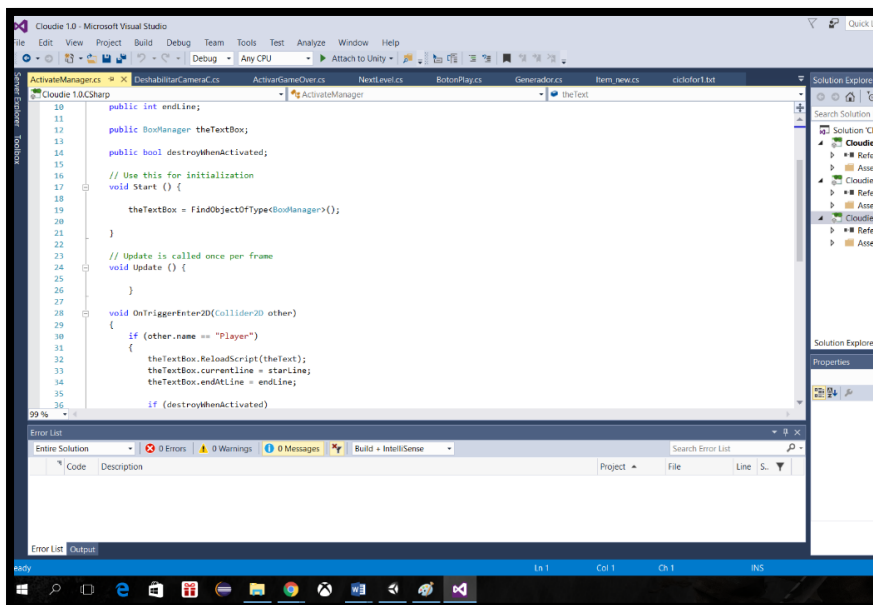
en todos los trabajos, por ejemplo, infraestructura, conexiones y comunicación

entre sucursales, se da gracias a las redes, una de las ramas de Ingeniería de sistemas, otro ejemplo, es el uso

de las páginas web, aplicaciones de facturación, aplicaciones móviles, se da gracias al desarrollo de software una de las ramas más relevantes e importantes de la



Fuente autor



Fuente autor

ingeniería de sistemas.

Sin embargo se establece que la ingeniería de sistemas abarca infinitudes de funciones, sin embargo el mundo de la programación y el aumento del uso del software es tan grande que se divide esta categoría y propiamente se le da el nombre a

Ingeniería de Software, el cual identifica el proyecto de software a desarrollar, se

implementa la arquitectura de software indicado para dicho proyecto y se realiza el desarrollo del mismo, de igual forma por medio de documentación se deja constancia de cada proceso y modelamiento de todo lo que compone la aplicación finalizada.

En este proyecto, en el cual se desarrolla un juego para niños de 5 a 12 años, en el que por medio de dinamismo y eventos didácticos se desea enseñar al niño sobre términos básicos de la programación. Para su desarrollo se evidencia en primer lugar los motores de video Juegos como lo es el Unity en su versión 5.2.2 y el entorno de programación Visual Studio 2012. El motor de video juegos Unity presenta varias propuestas para el desarrollo del juego, este juego PIPOO 1.0 está desarrollado en 2D, y de igual forma conjuntamente con la ingeniería de software se usa el lenguaje C# para su desarrollo y eventualmente términos de física como Rigidbody2D para establecer gravedad a los personajes del juego. De igual forma el Visual Studio 2012, el entorno en el cual se crea el código para darle vida al juego de PIPOO 1.0.

2.3.4. Lenguajes de programación e ingeniería de software

Los lenguajes de programación se clasifican por niveles, entre más cercanas sean las instrucciones al lenguaje del procesador se dice que es de bajo nivel, entre más abstracto sea el lenguaje se dice que es de alto nivel. El software está conformado por una serie de instrucciones y datos, que permiten la utilización de los recursos que un computador tiene, por lo tanto se puede resolver una gran cantidad de problemas. El software le da vida al computador, haciendo que cada componente funcione de forma correcta y organizada. Software del sistema, el cual es un conjunto de programas que administran recursos de una computadora, como por ejemplo, Unidad central del proceso y dispositivos periféricos y de comunicaciones. Software de aplicaciones, son programas que son escritos para los usuarios, para realizar una tarea específica, son aplicaciones como hojas de cálculo, o procesamiento de texto. De igual forma se encuentra el software desarrollado por empresas como por ejemplo ORACLE, VISUAL STUDIO, etc.

Software de usuario final es el que se encuentra sobre el software de aplicación y a través del software del sistema. Es decir usuarios que deben realizar queries o consultas en la base de datos usan por ejemplo ORACLE que es un software de aplicación que funciona sobre el software del sistema.

2.3.5. Lenguajes de animación y motores de videojuegos

Unity es un motor de Videojuegos ó también conocido como game engine. Es un ambiente de desarrollo que provee herramientas para crear casi cualquier tipo de videojuego de diversos géneros. Este game engine se ha convertido en uno de los más populares, y ha ganado mucho terreno en la escena Indie. Sus herramientas son fáciles de usar, y es muy versátil a la hora de exportar videojuegos a diferentes plataformas (Desde teléfonos inteligentes hasta consolas de última generación). Debido a su popularidad Unity cuenta con una amplia comunidad de desarrolladores que constantemente están enriqueciendo foros de consulta y solución de problemas, de la misma forma cuenta con una documentación muy precisa sobre cada uno de los features que contiene.

Dentro de las facilidades que Unity presenta podemos encontrar:

- Motor de renderizado.
- Motor de físicas y colisiones.
- Motor de audio.
- Herramientas de animación.
- Herramientas de networking para juegos multijugador y conexión con servicios externos.
- Compatibilidad con diferentes lenguajes de programación.

2.3.6. Construcción del marco conceptual

Se tendrá en cuenta los siguientes elementos para la consecución, construcción y representación de la aplicación:

2.4. Metas a alcanzar

- Metas a corto plazo: Se realizará la planeación del problema, las soluciones posibles y se detectaron variables de relevancia como: Personajes, historia, colores, puntuaciones, premios, escenarios.
- Metas a mediano plazo: Se desarrollará el planteamiento, se estructurará los escenarios, movimientos de personajes. Se realizará todas las acciones pertinentes de cada escena.
- Metas a largo plazo: Desarrollar el proyecto y realizar la entrega del mismo, con las opciones de pruebas de la aplicación.

2.5. Principios

El producto de software estará regido por los siguientes principios:

- Ambiente óptimo y de fácil manejo para la realización de acciones en la aplicación.
- El mantenimiento de la aplicación será sencilla, contará con actualizaciones para un módulo en específico sin que este afecte las demás escenas.
- La aplicación contará con seguridad integral, según su uso si es vía web o aplicación de escritorio, teniendo en cuenta los puertos de accesos.
- La interfaz se realizará a través de C# como lenguaje de programación principal, utilizando como entorno de programación Visual. Net y Unity como opción de motor de videojuegos.

2.5.1. Enfoque

El presente proyecto está destinado para los niños de edades de 5 a 12 años.

2.5.2. Productos del Modelo

El producto de software brindará los siguientes resultantes:

- Producto final: Software Educativo Orientado a la Web de enseñanza de programación
- Acceso y conexión con la Base de datos.
- Manuales de usuario y de operación.

2.5.3. Controles

El aplicativo contará con los siguientes controles:

- Registro de roles por usuario

2.5.4. Glosario

Lenguajes de programación: Un Lenguaje de programación es un conjunto de símbolos y caracteres combinados entre sí de acuerdo con una sintaxis establecida, que permite transmitir instrucciones a la CPU

Bases de Datos: Una base de datos es un sistema de archivos electrónico. Las bases de datos tradicionales se organizan por campos, registros y archivos.

Bases de datos Relacional: es una colección de elementos de datos organizados en un conjunto de tablas formalmente descritas desde la que se puede acceder a los datos o volver a montarlos de muchas maneras diferentes sin tener que reorganizar las tablas de la base.

C#: Es un lenguaje de programación implementado bajo el paradigma de programación orientada a objetos. Este lenguaje fue creado por Anders Hejlsberg (Mismo creador de Turbo Pascal y Delphi). C# es un lenguaje estrictamente tipado, y está basado en la creación, modificación y extensión de tipos, del inglés type.

Dato: Información concreta sobre hechos, elementos, etc., que permite estudiarlos, analizarlos o conocerlos.

Registro: Unidad completa de información que está asociada a un proceso de entrada o de salida.

SQL: (Structured Query Language) es un lenguaje de programación estándar e interactiva para la obtención de información desde una base de datos y para actualizarla. Aunque SQL es a la vez un ANSI y una norma ISO, muchos productos

de bases de datos soportan SQL con extensiones propietarias al lenguaje estándar.

Unity: es un motor de Videojuegos ó también conocido como game engine. Es un ambiente de desarrollo que provee herramientas para crear casi cualquier tipo de videojuego de diversos géneros. Este game engine se ha convertido en uno de los más populares, y ha ganado mucho terreno en la escena Indie. Sus herramientas son fáciles de usar, y es muy versátil a la hora de exportar videojuegos a diferentes plataformas (Desde teléfonos inteligentes hasta consolas de última generación).

Videojuegos: Los videojuegos son experiencias interactivas orientadas al entretenimiento que se ejecutan sobre plataformas electronicas. En la actualidad existen muchas definiciones sobre las que se pueden discutir, una de las más aceptadas a nivel global es la de Raph Koster, autor del libro Theory of Fun quien dice que un videojuego es: “An Interactive Experience that provides the player with an increasinly challenging sequence of patterns which he or she learns and eventually masters”.

Videojuegos Educativos: Los videojuegos educativos, también son experiencias interactivas que cumplen con la definición de videojuego, pero más que en entretener, su objetivo general es enseñar algún concepto o habilidad. En el contexto global los videojuegos son ampliamente aceptados como una herramienta de aprendizaje, debido a que su factor de diversión sirve como enganche para que las personas decidan utilizar el videojuego.

Animator: es el componente principal del sistema Mecanim de animación en Unity. Se representa por medio de máquinas de estado que definen los diferentes estados en los que puede estar un GameObject, las transiciones posibles entre dichos estados y parámetros.

Colisión: Una colisión, en el contexto de los videojuegos, es cualquier situación en la que 2 objetos están en contacto. Aunque en la vida real las personas tienden a pensar que las colisiones involucran el choque violento entre objetos que viajan a altas velocidades, en los videojuegos no es tan dramático. La misma lógica que resuelve las colisiones de objetos en reposo se usa para resolver las colisiones de objetos que viajan a alta velocidad. En el contexto de los videojuegos el término

colisión es sinónimo de contacto.

Collider: Un collider en Unity define la forma de la geometría que se va a usar para el cálculo de las colisiones de un objeto dentro del juego. Los cálculos de las colisiones son costosas en términos de operaciones del procesador, por esta razón se aconseja que la geometría de colisión sea sencilla.

FixedUpdate(): El método FixedUpdate se llama cada cierto tiempo, parecido al Update(), pero el FixedUpdate se llama siempre cada una cantidad fija de milisegundos que se puede configurar por el usuario. Debido a la forma en la cual se actualiza la física de un GameObject, el FixedUpdate es el método más apropiado para actualizar parámetros que tengan que ver con la física. Por ejemplo, cuando se añade una fuerza a un Rigidbody se debe hacer en el FixedUpdate, debido a que el método se llama antes de cada paso de la simulación física.

Box Collider: El box collider es un tipo específico de collider. En unity existen 2 Box Collider, un cuadrado si es el caso de 2 dimensiones y un cubo en el caso de 3 dimensiones. Los dos colliders tienen en común el material, y una variable que se encarga de saber si el collider es trigger, o no. El box collider de 3D es un cubo al que se le puede cambiar la posición del centro del cubo, para que el usuario seleccione la posición del collider según sus preferencias, de la misma forma, puede cambiar la escala de cada uno de las dimensiones del cubo.

Gizmo: En Unity el gizmo es una ayuda visual que sirve para modificar el componente transform de los objetos de juego. Los gizmos se ubican en la posición global del objeto y para modificarlos hay que seleccionar uno o más de los ejes que se desean transformar. En unity el gizmo de transformación tiene una representación diferente para cada uno de los componentes del transform (posición, rotación y escala).

Layout: Layout es una palabra que se usa en varios contextos con significados muy parecidos. En general Layout hace referencia a la organización de múltiples objetos dentro de un área o espacio. Por ejemplo en los deportes Layout hace referencia a la organización (coordenadas de posición) de los jugadores dentro de una estrategia. En el contexto de los programas de computadora el layout hace

referencia a cómo el usuario puede visualizar y modificar la distribución de los diferentes paneles de la interfaz gráfica (barras de herramientas, botones, espacios de trabajo, etc.).

Obstáculos: Los obstáculos, como los enemigos [Ver Enemigos en el glosario de la semana 1], son objetos que le impiden al jugador cumplir los objetivos del juego de manera lineal, y existen para que el jugador se vea obligado a adquirir una o más habilidades para completar el juego. Por lo general los objetos son perceptibles a los ojos del jugador y tienen forma física, como púas, rocas, trampas, lava, agua etc.

Recursos Gráficos: Los recursos gráficos hacen referencia a cualquier elemento visual que sea presentado durante la ejecución del videojuego. Algunos ejemplos de recursos gráficos en el demo 2D sobre Unity son los siguientes:

- Sprites para personajes.
- Texturas para el fondo de la escena.
- Elementos de la interfaz gráfica del usuario.
- Sprites para elementos ornamentales.
- Texturas para partículas como humo.

Sprite: Un sprite en los videojuegos es una imagen 2D que representa gráficamente un gameobject. Los sprites en Unity pueden ser sencillos o contener una animación. Desde la versión 4.3 de Unity existe un nuevo tipo de textura llamado Sprite, la cual se puede utilizar para dibujar personajes, objetos de adorno, trampas, paredes, etc. También un Sprite permite modificar algunas características muy útiles a la hora de desarrollar videojuegos, por ejemplo:

- El punto central de la imagen o pivote.
- Los límites de la imagen.
- La textura del sprite.

Unity permite definir la imagen de un sprite a partir de un atlas, para que funcione debe especificar la sección del atlas que representa el Sprite.

Sprite Editor: Es un panel Unity que está disponible para las texturas de tipo sprite que selecciona como “multiple” su sprite type. En el sprite Editor se puede “cortar”

y separar las imágenes dentro de un Atlas [ver atlas en glosario semana 1], para obtener múltiples imágenes por separado. El editor permite configurar la forma y tamaño de las sub imágenes del atlas.

El Sprite Editor es una ventana flexible y que sirve de ayuda para configurar imágenes irregulares de formas y tamaños distintos, como es el caso de los elementos que componen un mismo personaje; de la misma forma sirve para imágenes ordenadas y regulares como es el caso de una animación, cuya distribución es regular con respecto al tamaño y forma de cada uno de los frames que componen la animación

Videojuegos: Los videojuegos son experiencias interactivas orientadas al entretenimiento que se ejecutan sobre plataformas electrónicas. En la actualidad existen muchas definiciones sobre las que se pueden discutir, una de las más aceptadas a nivel global es la de Raph Koster, autor del libro Theory of Fun quién dice que un videojuego es: “An Interactive Experience that provides the player with an increasingly challenging sequence of patterns which he or she learns and eventually masters”.

Videojuegos Educativos: Los videojuegos educativos, también son experiencias interactivas que cumplen con la definición de videojuego, pero más que en entretener, su objetivo general es enseñar algún concepto o habilidad.

En el contexto global los videojuegos son ampliamente aceptados como una herramienta de aprendizaje, debido a que su factor de diversión sirve como enganche para que las personas decidan utilizar el videojuego.

3. DISEÑO METODOLOGICO

3.1 Tipo de Investigación

El tipo de investigación es básica y aplicada. Es básica, ya que emerge con un amplio marco teórico y la adquisición de conocimientos sobre la mejora del desarrollo cognitivo de los niños al utilizar juegos educativos. Es aplicada y así mismo se encuentra bastante vinculada con el tipo de investigación básica, al destacar los conocimientos mediante resultados. Con estos resultados se determinan los avances y así mismo el surgimiento de nuevos conocimientos y fortalecimiento de los actuales.

3.2. Metodología Seleccionada

La metodología seleccionada es con base a la metodología ISE (ingeniería de software educativo) de Álvaro Galvis, en el cual sigue el proceso de análisis, diseño, desarrollo, entorno de pruebas y pruebas de campo. Con base a lo anterior, se utilizara dicha metodología, ya que *“La metodología de ISE.... ofrece mecanismos de análisis, diseño, educativo y comunicacional, prueba piloto y de campo bastante sólidos, toda vez que se fundamentan en principios educativos, comunicacionales y de tecnología educativa de validez comprobada.”* No solo se realizara la metodología por enfoque educativo, sino que al mismo tiempo esta metodología se acopla al ciclo de vida de un software (análisis, diseño y desarrollo)

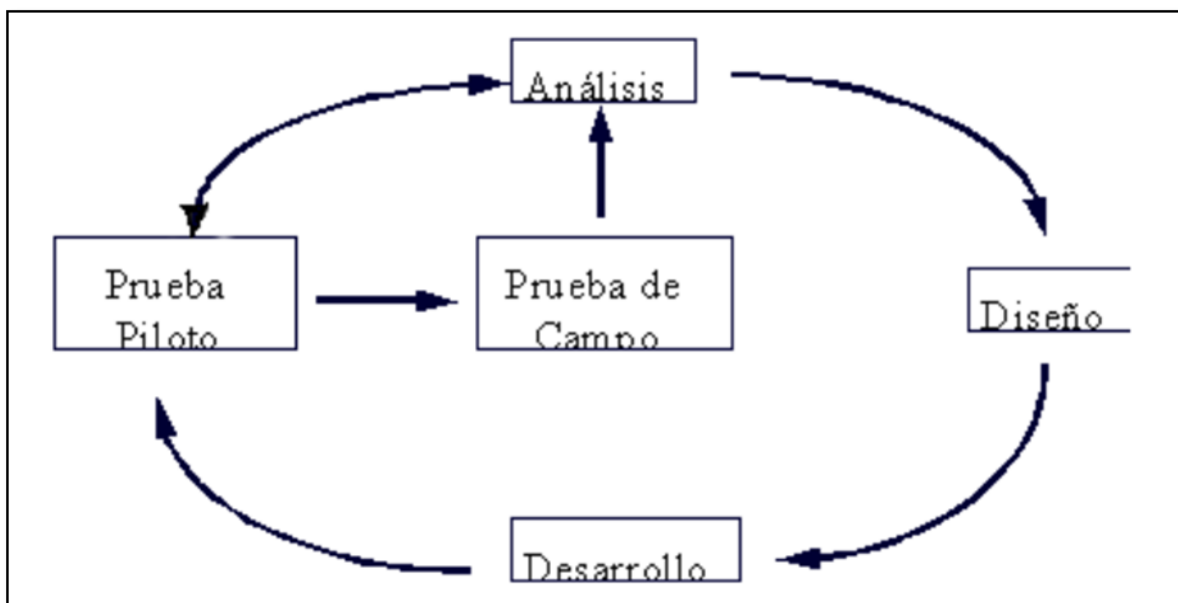


Ilustración 3 – Metodología propuesta por Álvaro Galvis
http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-106223_archivo.pdf

3.2.1. Análisis de Necesidades

Se verifica las necesidades del sistema, enmarcando la metodología para el desarrollo del juego y analizar las propiedades necesarias para el avance del mismo. Las propiedades a detallar serán el manejo y el proceso determinado en el juego.

Población objetivo: Las características de la población objetivo dentro de las cuales están enmarcadas este proyecto tenemos:

- Los niños y niñas entre los 5 a 12 años podrán utilizar el juego para optimizar sus conocimientos. Ya que es un juego implementado para los niños de una forma didáctica para su aprendizaje.

3.3. Requerimientos del nuevo sistema

3.3.1. Requerimientos funcionales

Los requerimientos funcionales del proyecto son los siguientes:

- El sistema debe permitir mostrar cuadros de diálogos, para brindar una continuidad con el juego.
- El sistema debe permitir el almacenamiento de puntaje en los niveles de los niños entre las edades de 8 a 12 años.
- El sistema debe permitir escoger un personaje en las edades de 8 a 12 años.
- El sistema debe permitir mostrar la cantidad de puntaje que se recoge al escoger el objeto correcto, para los niveles de las edades de 5 a 8 años.
- El sistema debe permitir mostrar el código de la captura de un objeto determinado.
- El sistema debe mostrar mediante un mensaje emergente que se ha ganado la partida.
- El sistema debe permitir escoger los niveles para las edades entre los 5 a 8 años y los 8 a 12 años.

3.3.2. Requerimientos no funcionales

Los requerimientos NO funcionales del proyecto son los siguientes:

Seguridad: El Juego Pipoo se encuentra disponible mediante un servidor web y el hosting gratuito que otorga la página oficial de Unity para visualizar el juego:

- El sistema permitirá la creación de usuario de login, para los niveles para los niños de 8 a 12 años.
- El usuario al ingresar únicamente tendrá acceso al desarrollo de tres niveles para niños de 8 a 12 años.
- Se controlará el estado de los puertos, ya que estos deben estar cerrados, para que intrusos no puedan ingresar. De tal modo los puertos: 443 (HTTPS)

servidor web seguro, 3306(MYSQL) Base de datos. 80(HTTP) servidor web, 23(Telnet) acceso remoto, permanecerán en constante monitoreo.

Disponibilidad: El sistema estará disponible el 100% durante el día. Se dispondrá un 3% del tiempo en actualización de la base de datos o de versión, es decir que durante ese proceso el sistema estará disponible el 97% en el día. Sin embargo, cabe resaltar, que se pretende que dichas actualizaciones se ejecuten en horarios que no afecten el uso del juego.

Mantenibilidad: Por seguridad se realizará ventanas de mantenimiento al sistema, por parte de los desarrolladores a cargo. Dichos mantenimientos se realizarán una vez a la semana, en jornadas nocturnas, para no alterar el uso del juego PIPOO por parte del usuario.

Portabilidad: El sistema en el caso de requerir un traslado de plataforma o entorno será necesario tener en cuenta que:

- Se mantendrá el lenguaje de desarrollo C#, para el correcto y continuo funcionamiento.
- La plataforma de desarrollo será Visual Studio y Unity 5.2
- El sistema operativo para el continuo funcionamiento será Windows
- Es un juego que puede ser usado en los navegadores: Chrome, Mozilla, Opera, Safari.

3.4. Diseño del nuevo sistema

El juego consta de 5 niveles, divididos en: 2 niveles para niños de 5 a 8 años y 3 niveles de 8 a 12 años.

Niveles (5 a 8 años)

1: Escena I (primer nivel): Este nivel permite al usuario interactuar con objetos flotantes. Estos objetos son solicitados por el sistema, para que el personaje Pipoo

los encuentre y se visualice como se crea un objeto en la área de la programación.

2: Escena II (segundo Nivel): este segundo nivel es representado por el personaje Pipoo que se mantiene en movimiento y al atrapar objetos, se evidencia mediante mensajes la creación de ciclos sencillos.

Niveles (8 a 12 años)

- 1: Primer nivel del juego: Se evidencia un camino, el cual es recorrido por un personaje, el cual se puede escoger al inicio del juego. El camino será correctamente completado al indicar los pasos correctos al llegar al otro extremo.
- 2: Según nivel del juego: permite realizar un recorrido parecido al primero, pero es un camino más grande y complejo.
- 3: Tercer nivel del juego: permite realizar nuevamente un recorrido, aunque el diseño del camino es diferente a los dos anteriores. Se utilizan combos para usar símbolos y hacer que el personaje se mueva según lo ingresado.

3.4.1. Diseño mapa de navegación

A continuación se observa el mapa de navegación, que contiene el juego, para el desarrollo de los niveles. Se identifica la interacción entre el sistema y el usuario e incluso las múltiples opciones de juego que ofrece.

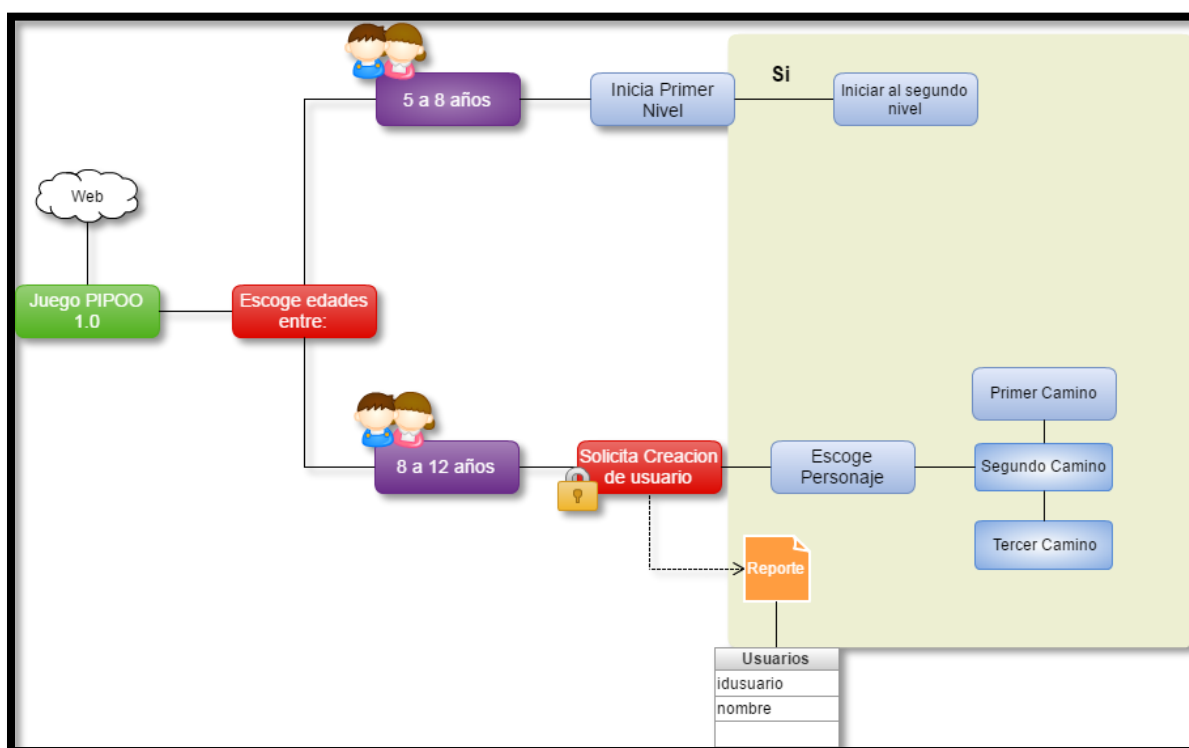


Ilustración 4 Mapa de Navegación- Fuente: Autor

Se podrá visualizar el juego vía web, desde cualquier navegador. Al generar el ingreso el juego le pedirá escoger las edades del usuario, según cada rango de edad entre 5 a 12 años, se dispondrá de niveles diferentes, el cual le enseñara al usuario a crear objetos y ciclos sencillos e incluso el seguimiento de secuencias, mediante órdenes. En el módulo de 8 a 12 años, se realizara la creación de usuario y se podrá escoger un personaje el cual acompañara al usuario en cada recorrido de los niveles.

3.4.2. Diseño orientado a objetos

En este segmento se identificara las interacciones que se crean desde el usuario con el sistema. Se determina lo anterior mediante una representación de diagramas UML con un enfoque informático.

3.4.3. Diseño de casos de uso

Los diagramas a tener en cuenta para el desarrollo del juego son los siguientes:

Descripción grafica de casos de usos en la interacción jugador e implementaciones del juego.

Caso de uso general

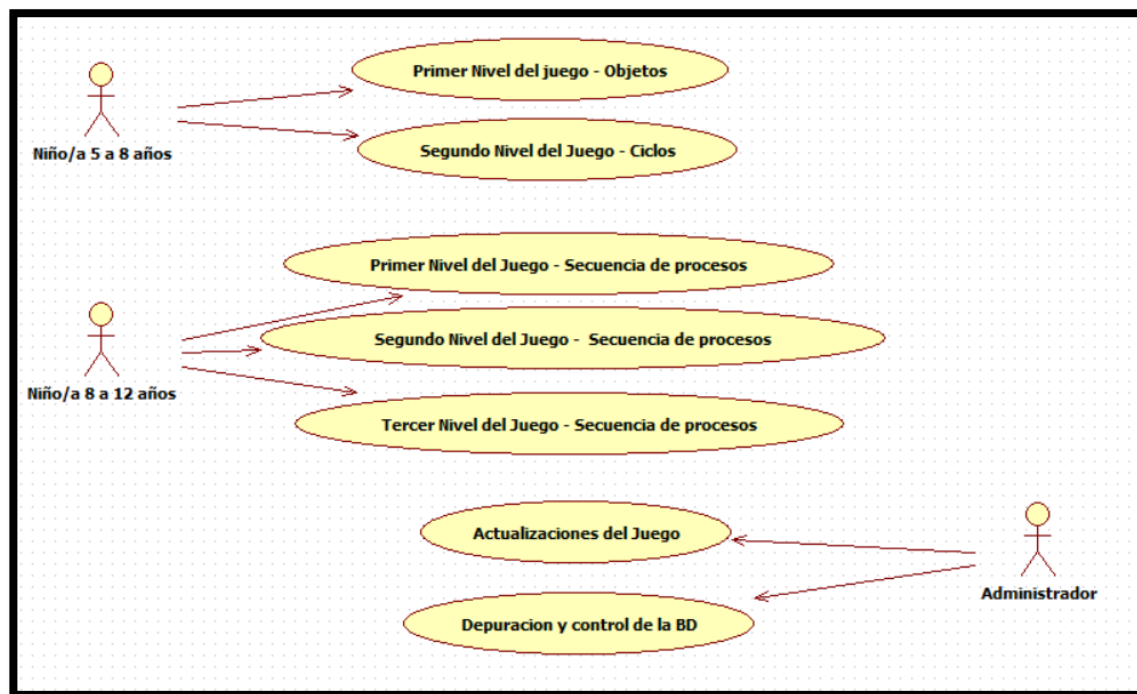


Ilustración 5 Caso de uso general - Fuente autor

Formato de Caso de Uso General	
Nombre	Caso de Uso general
Autor	Orlando Molina – Luisa Martínez
Fecha	15/01/2017
Descripción	Primera Interacción usuario - sistema
Actores	Usuario – Administrador
Precondiciones	El usuario con edades de 8 a 12 años debe estar autenticado en el sistema.

Flujo Normal	<ol style="list-style-type: none"> 1. El usuario escoge las opciones de juego según la edad que tenga. 2. Según la edad el usuario puede interactuar con: <ul style="list-style-type: none"> - Nivel 1 (5 a 8 años) - Nivel 2 (5 a 8 años) - Nivel 1 (8 a 12 años) - Nivel 2 (8 a 12 años) - Nivel 3 (8 a 12 años) 3. El sistema comprueba usuarios, almacena información de los mismos.
Flujo Alternativo	<ol style="list-style-type: none"> 4. El juego permite completar el nivel y da la opción para pasar de la siguiente, de igual forma por medio de un menú también se puede escoger el nivel. Muestra mediante un mensaje emergente cuando la respuesta sea correcta o es ganador de un nivel.
Pos condiciones	<p>En caso de equivocarse el usuario en alguna secuencia, genera un mensaje emergente de “has perdido”, de igual forma la secuencia permanece guardada para que el usuario continúe con el recorrido. (niveles de niños de 8 a 12 años)</p> <p>Fuente: Propia</p>

Caso de uso Inicio de sesión niños de 8 a 12 años

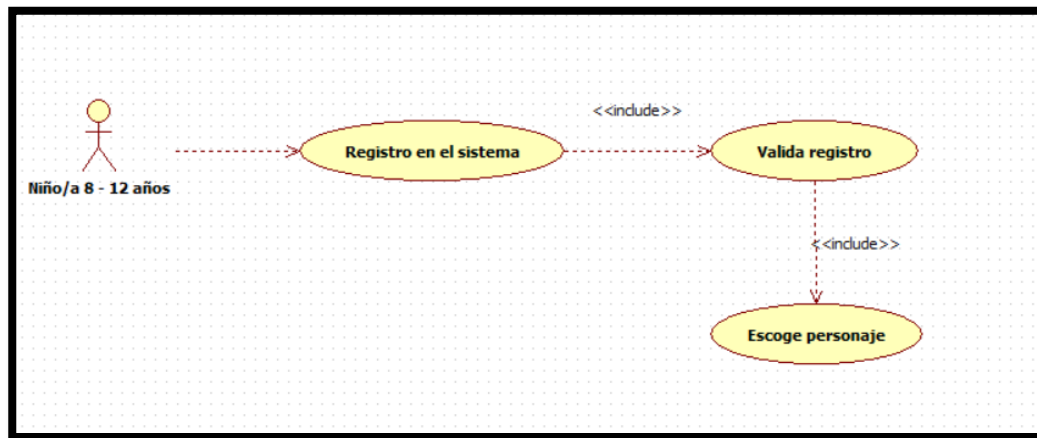


Ilustración 6 Caso de uso Inicio de sesión niños de 8 a 12 años - Fuente autor

Caso de Uso Inicio de sesión niños de 8 a 12 años

Nombre	Inicio de sesión niños de 8 a 12 años
Autor	Orlando Molina – Luisa Martínez
Fecha	15/01/2017
Descripción	Permite iniciar la sesión de usuario.
Actores	Usuario – 8 a 12 años
Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo Normal	<ol style="list-style-type: none">1. El actor pulsa sobre el botón para iniciar la sesión.2. El actor introduce los datos para realizar la validación en el sistema.3. El sistema verifica los datos ingresados, dicha información es almacenada en el sistema y permite elegir personaje al actor.
Flujo Alternativo	<ol style="list-style-type: none">4. El sistema verifica los datos ingresados, si la información es incorrecta, el sistema genera una ventana emergente solicitando nuevamente el ingreso de los mismos.
Pos condiciones	Los datos han sido almacenados en el sistema.

Fuente: Propia

Caso de uso Nivel I 5 a 8 años

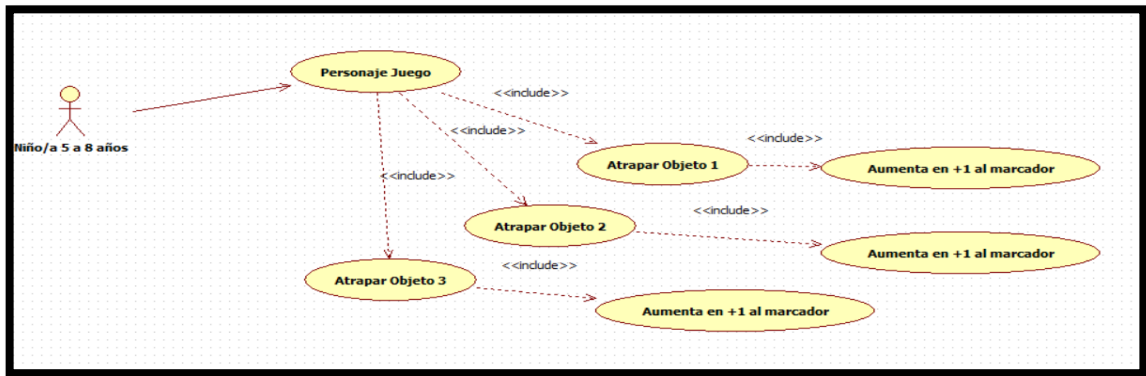


Ilustración 7 Caso de uso Nivel I 5 a 8 años - Fuente autor

Caso de Uso Nivel I 5 a 8 años

Nombre	Nivel I 5 a 8 años
Autor	Orlando Molina – Luisa Martínez
Fecha	15/01/2017
Descripción	Permite al actor interactuar con el primer nivel del juego.
Actores	Usuario – 5 a 8 años
Precondiciones	El usuario debe escoger la opción “edades 5 a 8 años”.
Flujo Normal	<ol style="list-style-type: none"> 1. El actor mediante las teclas de la computadora, desplaza el movimiento del personaje. 2. El actor mediante los movimientos y mensajes de ayuda alcanza el objeto a encontrar.
Flujo Alternativo	<ol style="list-style-type: none"> 3. El sistema mostrara por medio de mensajes que el objeto seleccionado es incorrecto.
Pos condiciones	El puntaje aumenta +1 cuando se escoge el objeto correcto.

Fuente: Propia

Caso de uso Nivel II 5 a 8 años

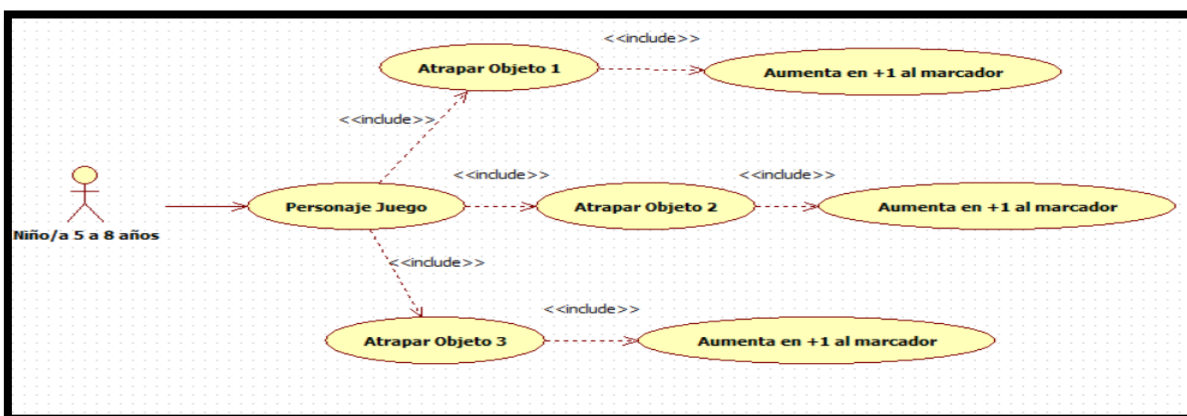


Ilustración 8 Caso de uso Nivel II 5 a 8 años Fuente autor

Caso de Uso Nivel I 5 a 8 años

Nombre	Nivel I 5 a 8 años
Autor	Orlando Molina – Luisa Martínez
Fecha	15/01/2017
Descripción	Permite al actor interactuar con el primer nivel del juego.
Actores	Usuario – 5 a 8 años
Precondiciones	El usuario debe escoger la opción “edades 5 a 8 años”.
Flujo Normal	<ol style="list-style-type: none"> 1. El actor mediante las teclas de la computadora, desplaza el movimiento del personaje. 2. El actor mediante los movimientos y mensajes de ayuda alcanza el objeto a encontrar.
Flujo Alternativo	<ol style="list-style-type: none"> 3. El sistema mostrara por medio de mensajes que el objeto seleccionado es incorrecto.
Pos condiciones	El puntaje aumenta +1 cuando se escoge el objeto correcto.

Fuente: Propia

Caso de Uso Niveles 8 a 12 años

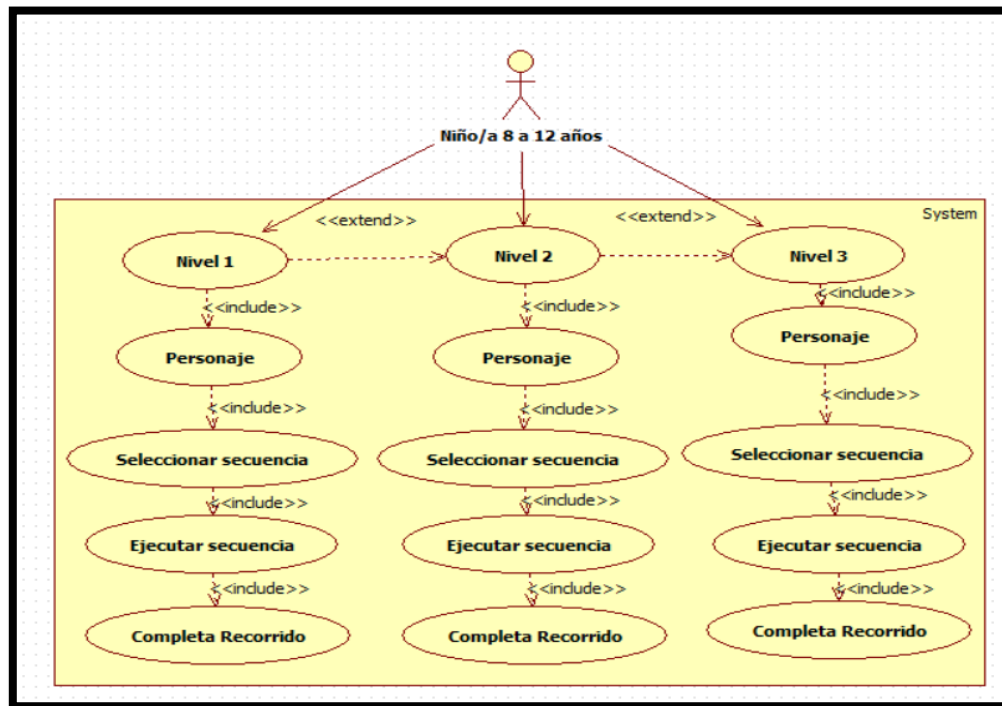


Ilustración 9 Caso de Uso Niveles 8 a 12 años Fuente autor

Caso de uso Niveles 8 a 12 años

Nombre	Niveles 8 a 12 años
Autor	Orlando Molina – Luisa Martínez
Fecha	15/01/2017
Descripción	Permite al actor interactuar con el primer nivel del juego.
Actores	Usuario – 8 a 12 años
Precondiciones	El usuario debe autenticarse en el sistema
Flujo Normal	<ol style="list-style-type: none"> 1. El actor escoge el nivel el cual va a jugar. 2. El actor mediante secuencias, registradas como botones de izquierda, derecha, arriba y abajo, crea el recorrido que el personaje debe hacer para llegar al final del camino.

Flujo Alternativo	3. El sistema mostrara un mensaje emergente de “perdió” cuando la secuencia es incorrecta, permanece visible el camino correctamente registrado.
Pos condiciones	El sistema almacena puntaje registrado.
Fuente: Propia	

3.4.4. Diagrama de clases

Descripción grafica de las clases de estructura sistema modelado mediante sus atributos, operaciones (o métodos), las relaciones entre los objetos y eventos durante la ejecución del juego.

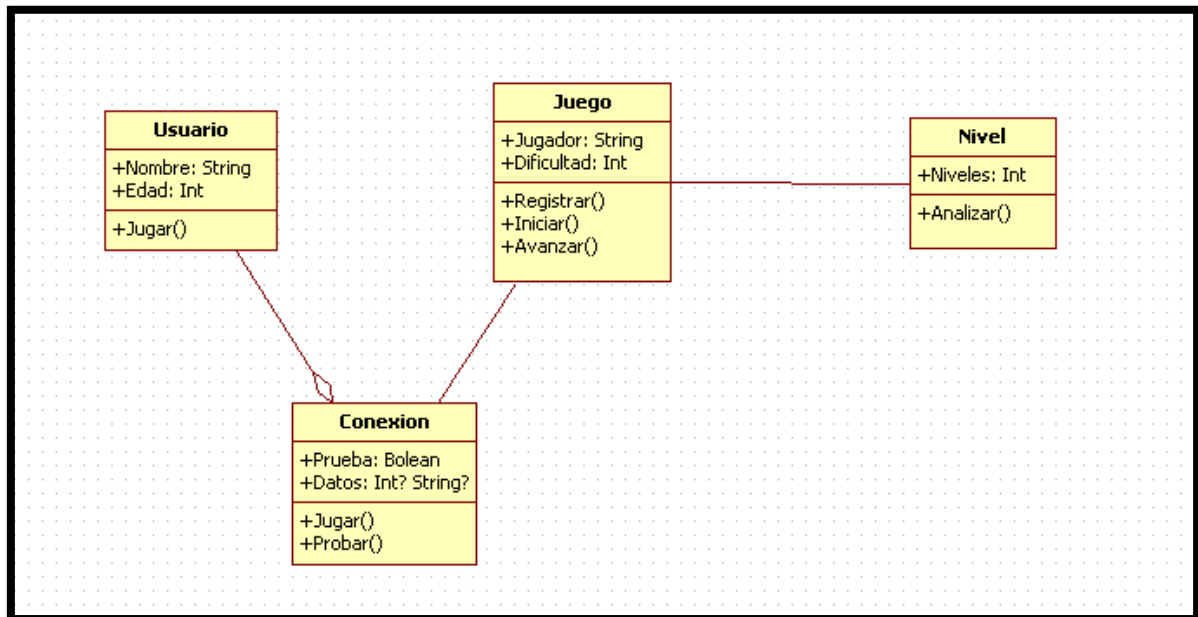


Ilustración 10 Diagrama de Clases Fuente autor

3.4.5. Diagrama de secuencia

Diagrama de interacción para los eventos específicos que se ejecutan en el juego en el que se muestra los objetos como líneas de vida y tiempo.

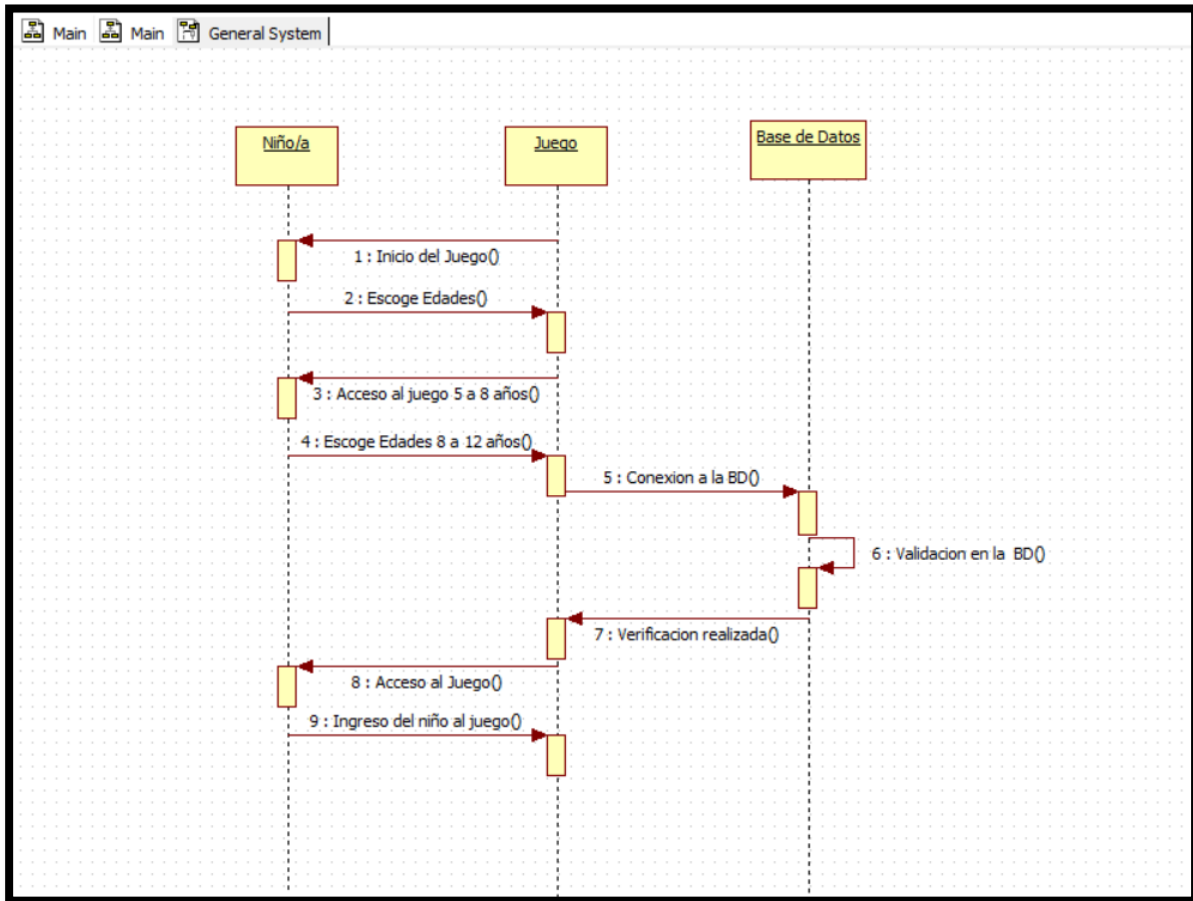


Ilustración 11 Diagrama de Secuencia - Fuente autor

Diagrama Juego 5 a 8 años Nivel I y Nivel II

Diagrama de secuencia para el evento en los niños entre 5 a 8 años con las líneas de vida y tiempo vinculados al rango de edad y su interacción con los niveles de dificultad para el primer nivel.

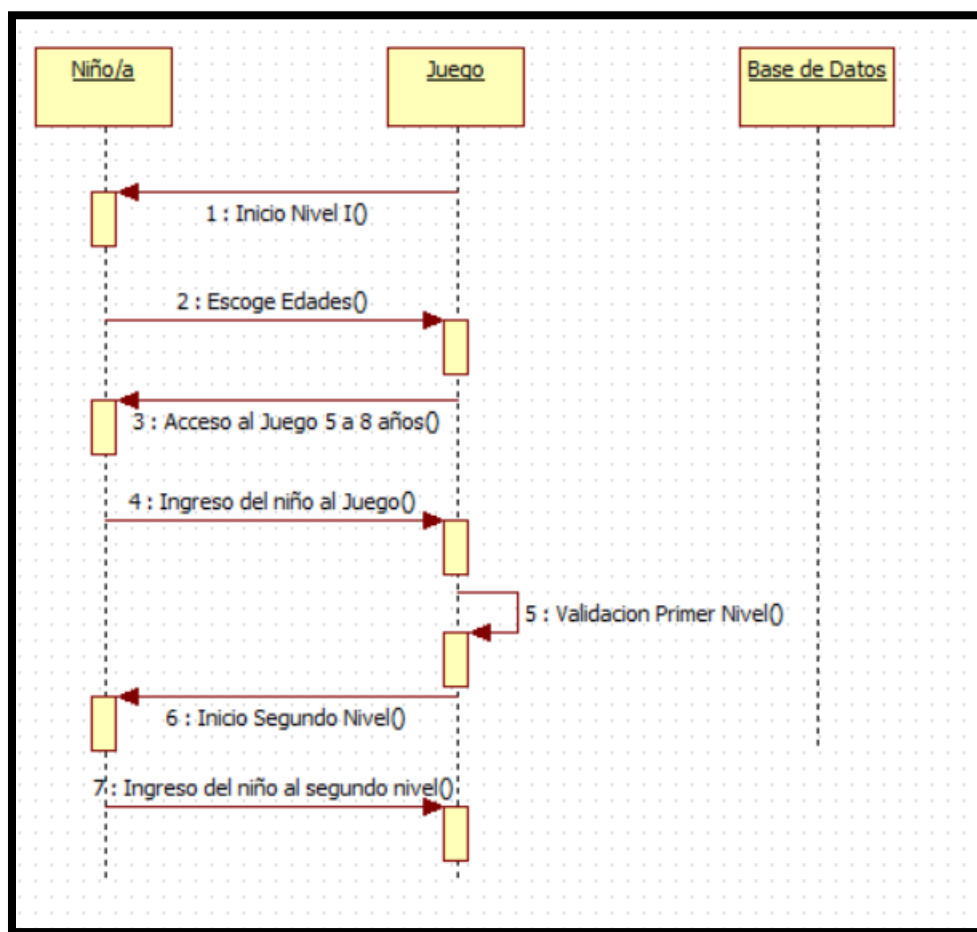


Ilustración 12 Diagrama Juego 5 a 8 años Nivel I y Nivel II - Fuente autor

Diagrama Juego Nivel 1 8 a 12 años

Diagrama de secuencia para el evento en los niños entre 8 a 12 años con las líneas de vida y tiempo vinculados al rango de edad y su interacción con los niveles de dificultad para el primer nivel.

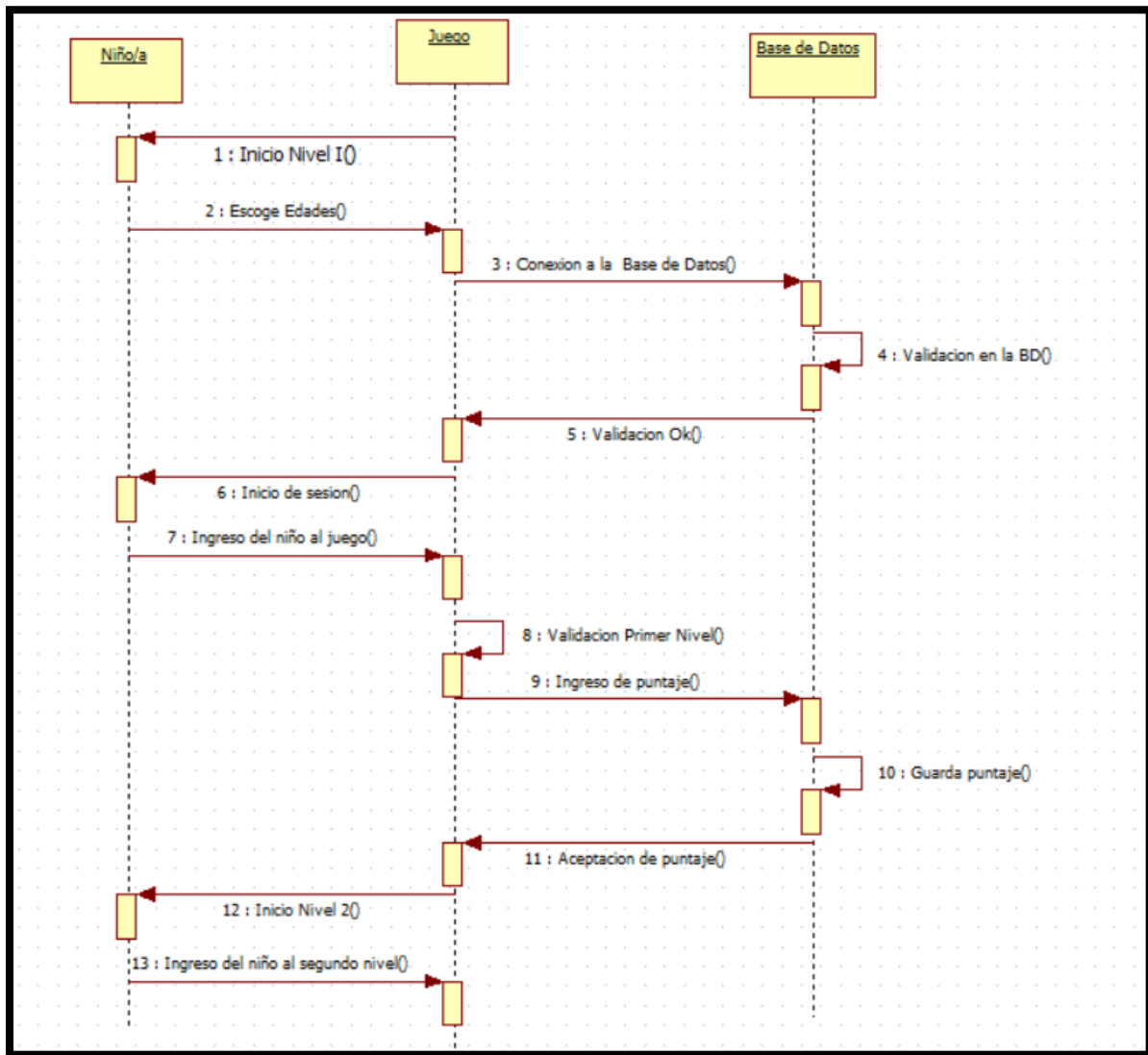


Ilustración 13 Diagrama Juego Nivel 1 8 a 12 años - Fuente autor

Diagrama Juego Nivel 2 8 a 12 años

Diagrama de secuencia para el evento en los niños entre 8 a 12 años con las líneas de vida y tiempo vinculados al rango de edad y su interacción con los niveles de dificultad para el segundo nivel.

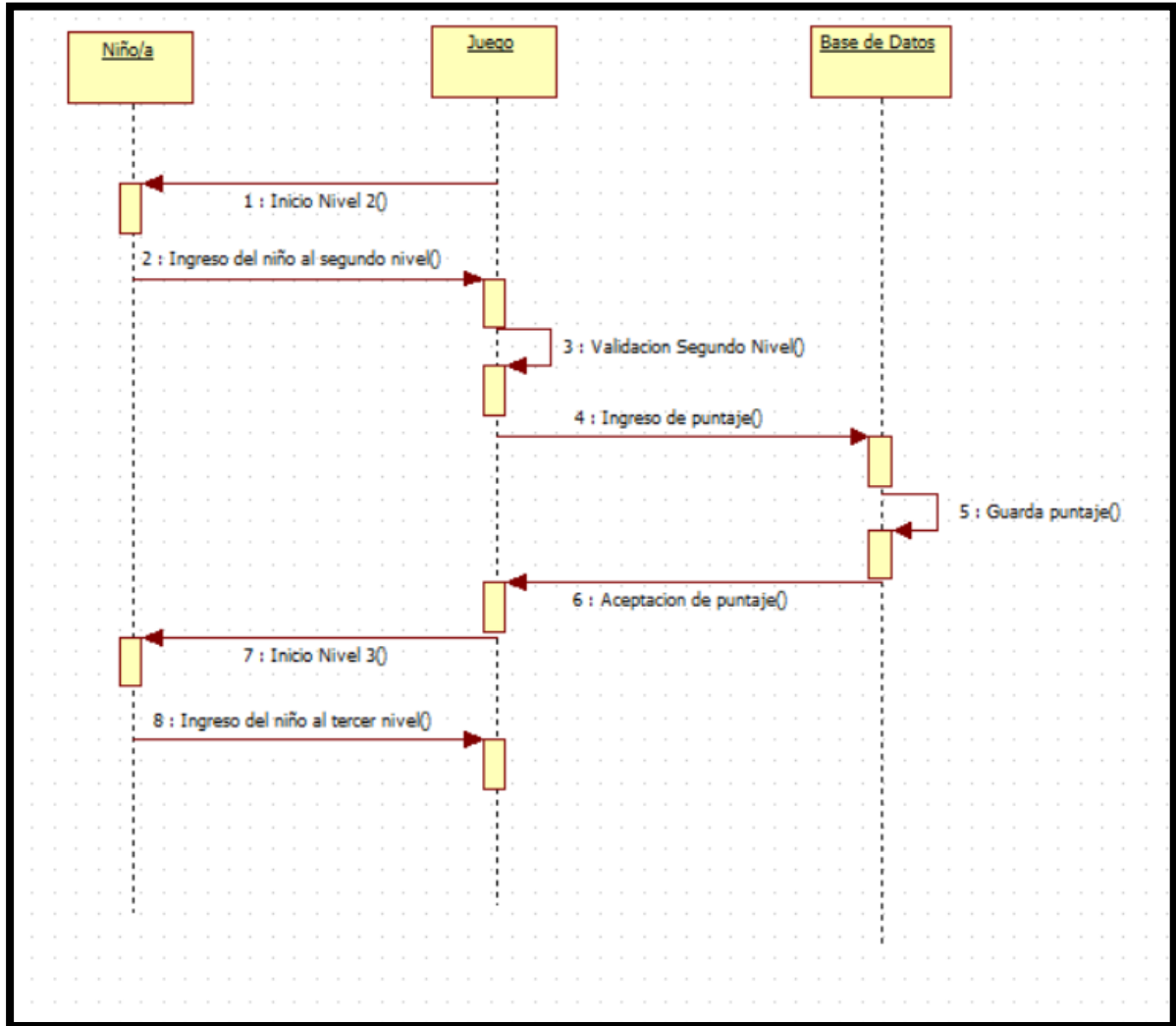


Ilustración 14 Diagrama Juego Nivel 2 8 a 12 años - Fuente autor

Diagrama Nivel 3 8 a 12 años

Diagrama de secuencia para el evento en los niños entre 8 a 12 años con las líneas de vida y tiempo vinculados al rango de edad y su interacción con los niveles de dificultad para el nivel final.

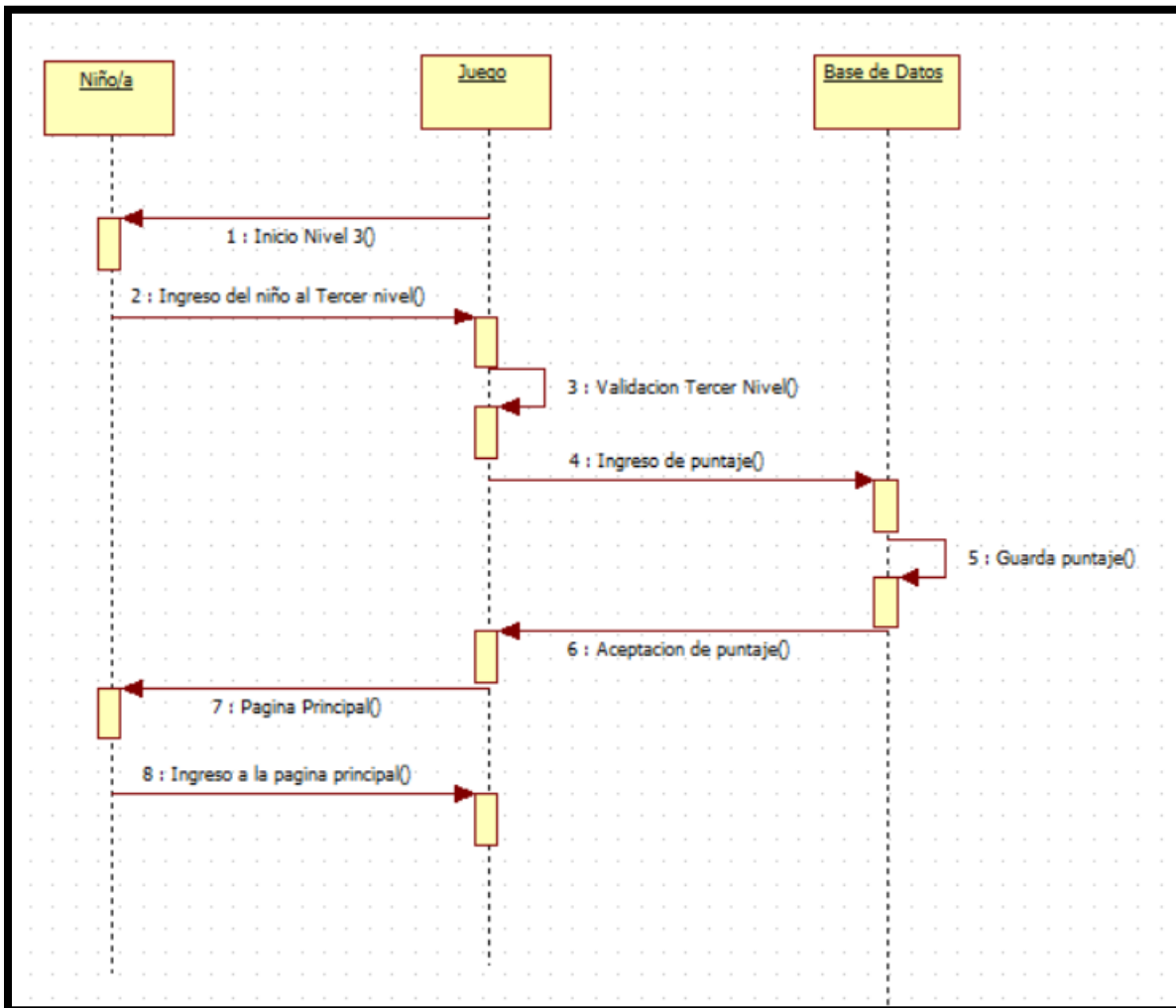


Ilustración 15 Diagrama Nivel 3 8 a 12 años - Fuente autor

3.4.6. Diagrama de Actividades

Descripción de actividades durante el proceso de juego para los diferentes niveles para el usuario.

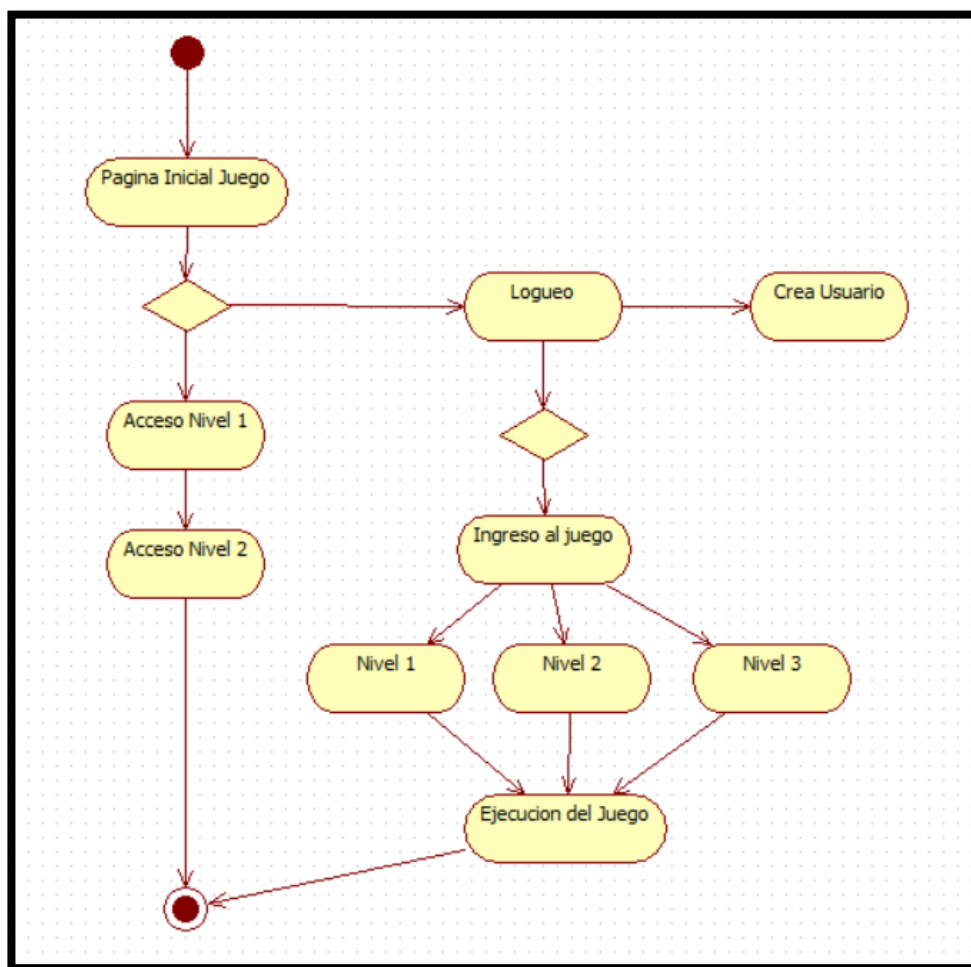


Ilustración 16 Diagrama de Actividades -Fuente autor

3.4.7. Diagrama de Paquetes

Descripción grafica de los paquetes que poseen los diferentes objetos durante el procesos de juego clasificados según su funcionalidad y objetivo.

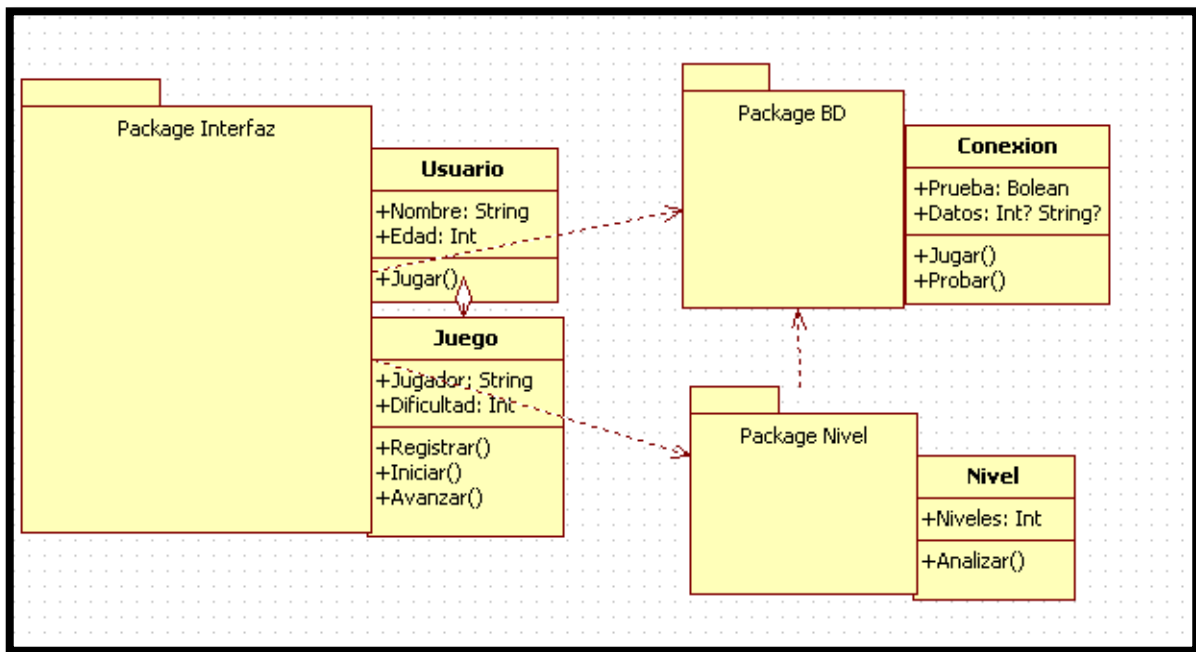


Ilustración 17 Diagrama de Paquetes - Fuente autor

3.4.8. Diagrama de Componentes

Descripción de los componentes que a su vez componen clases y objetos que interactúan en una interfaz gráfica para el usuario con una lógica de negocio (juego).

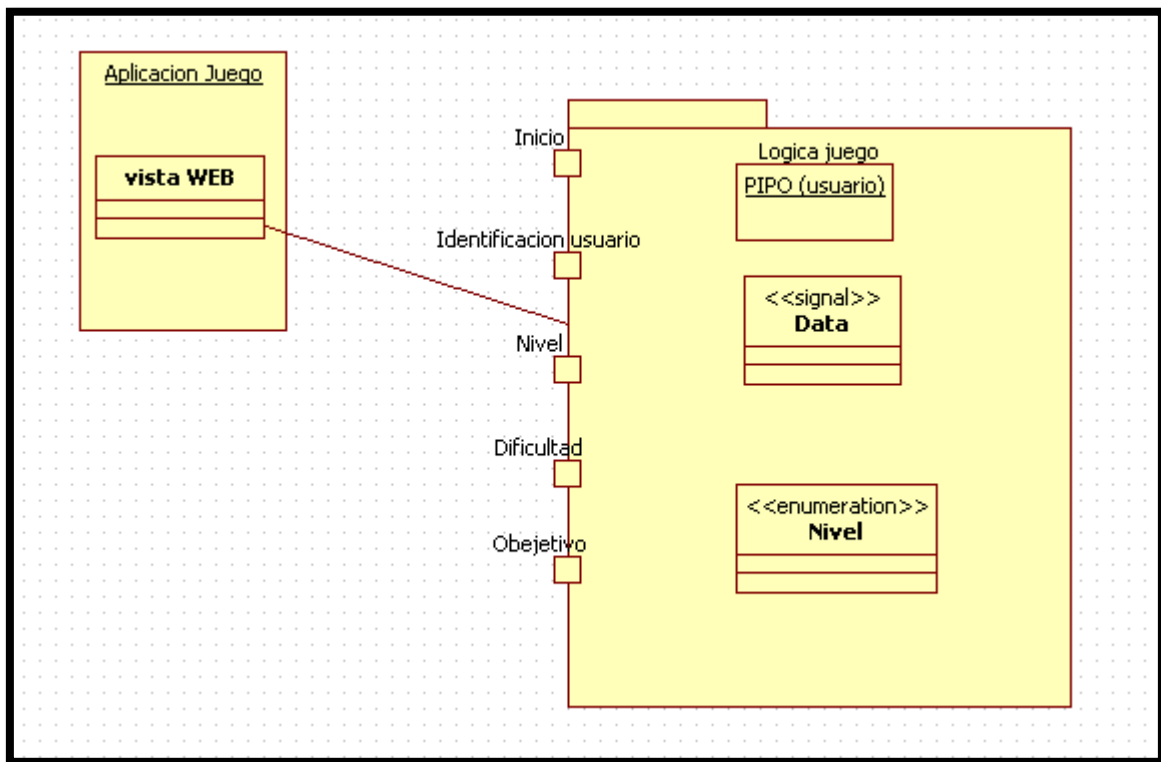


Ilustración 18 Diagrama de Componentes - Fuente autor

3.4.9. Diagrama de Despliegue

Despliegue de actividades en todos los elementos que componen el sistema e intercambian atributos e información para llegar al objetivo del juego.

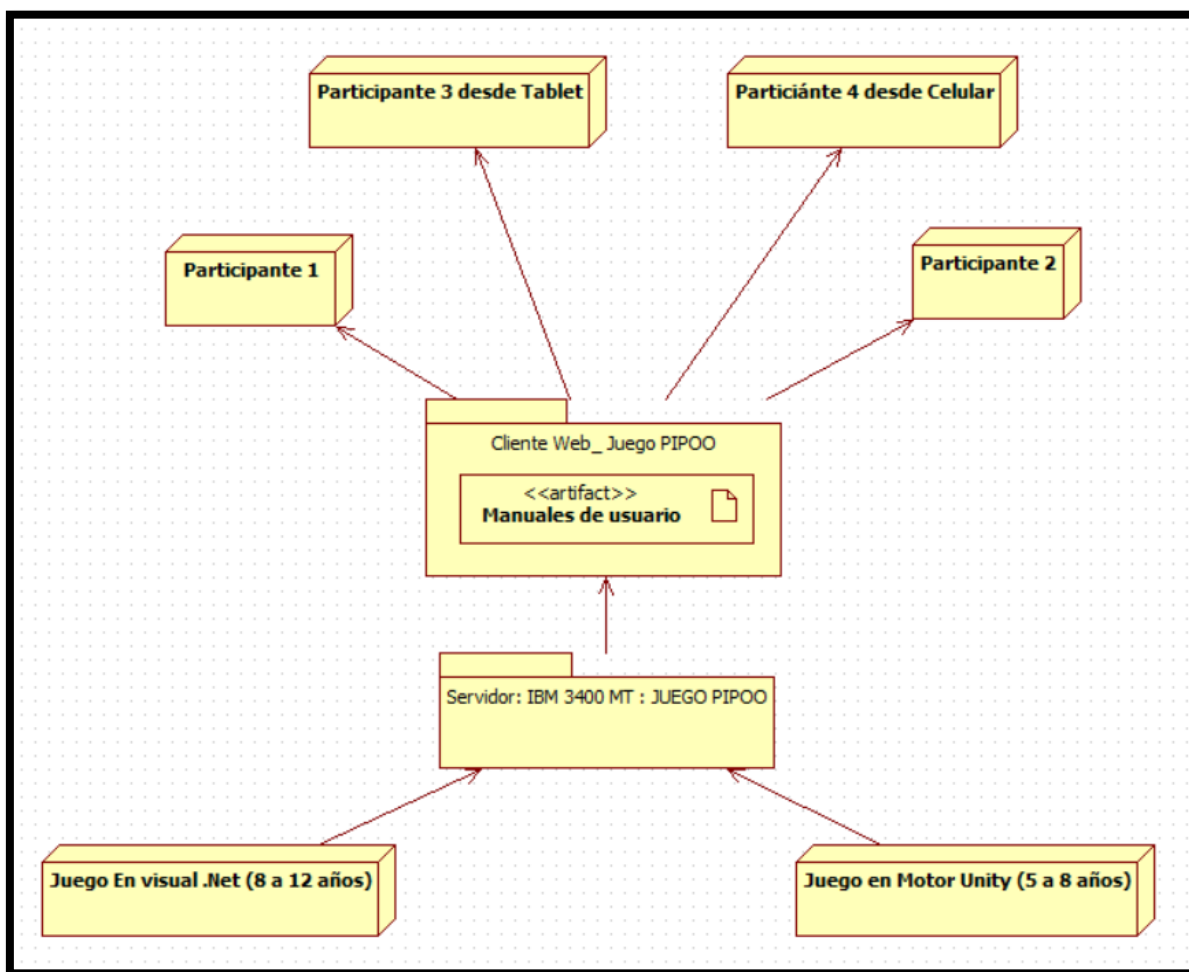
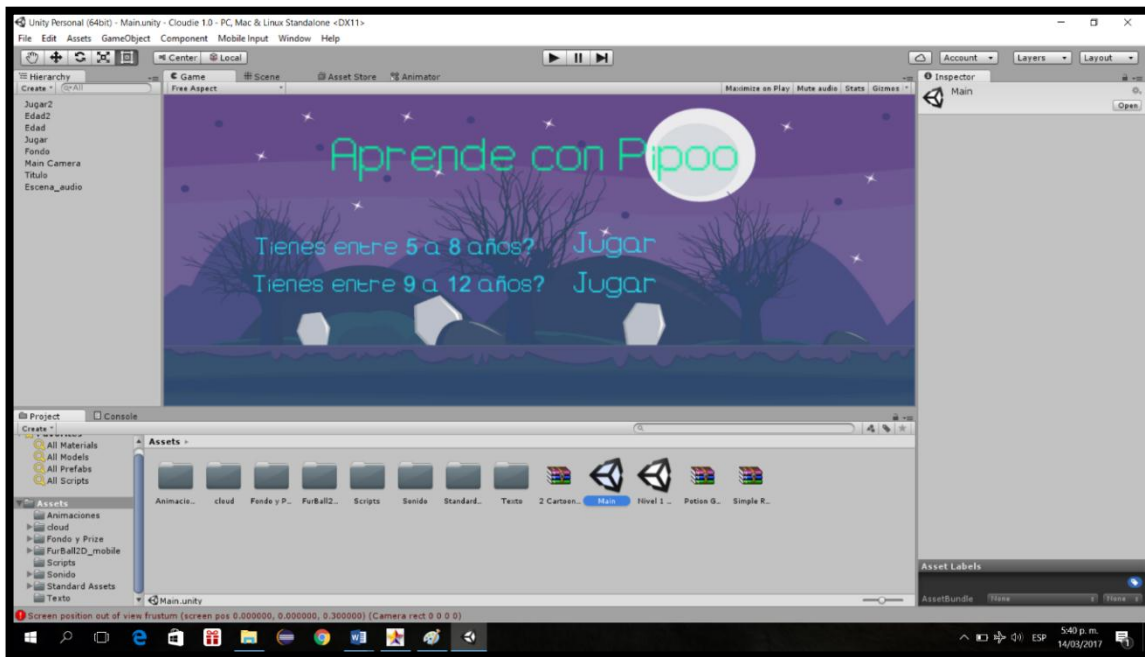


Ilustración 19 Diagrama de Despliegue – Fuente autor

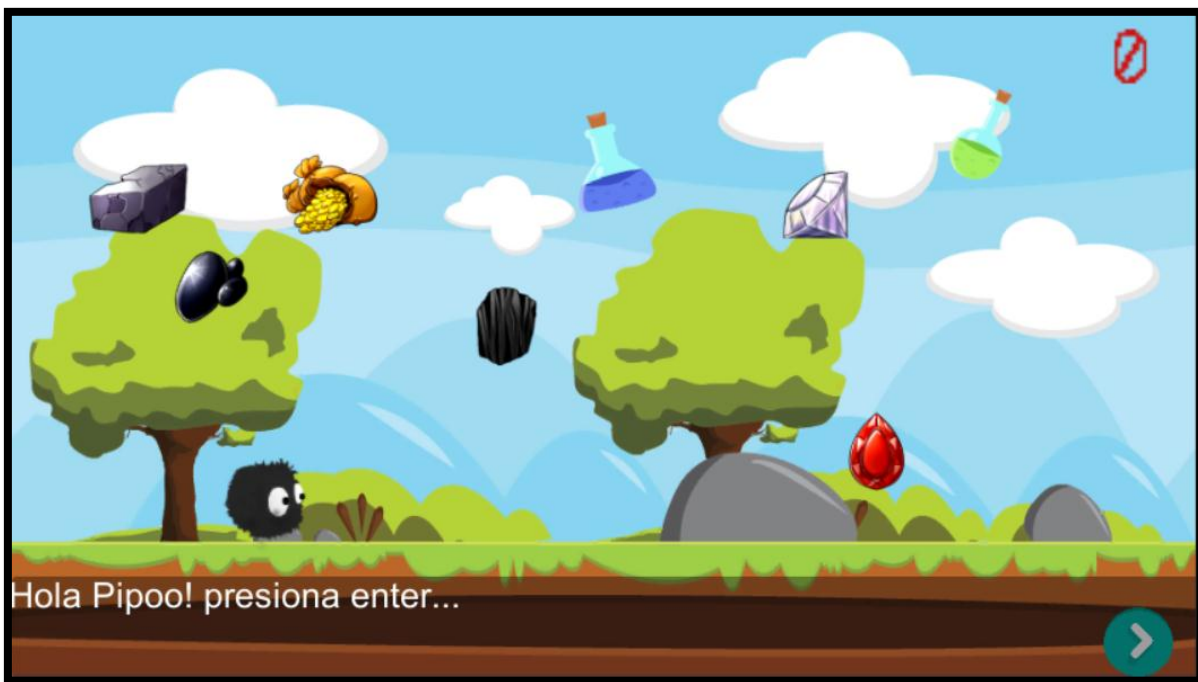
3.5. Diseño de interface de usuario

A continuación se observara el desarrollo del juego mediante la interfaz que es visualizada por el usuario.



Fuente Autor

corresponde al primer nivel del juego:



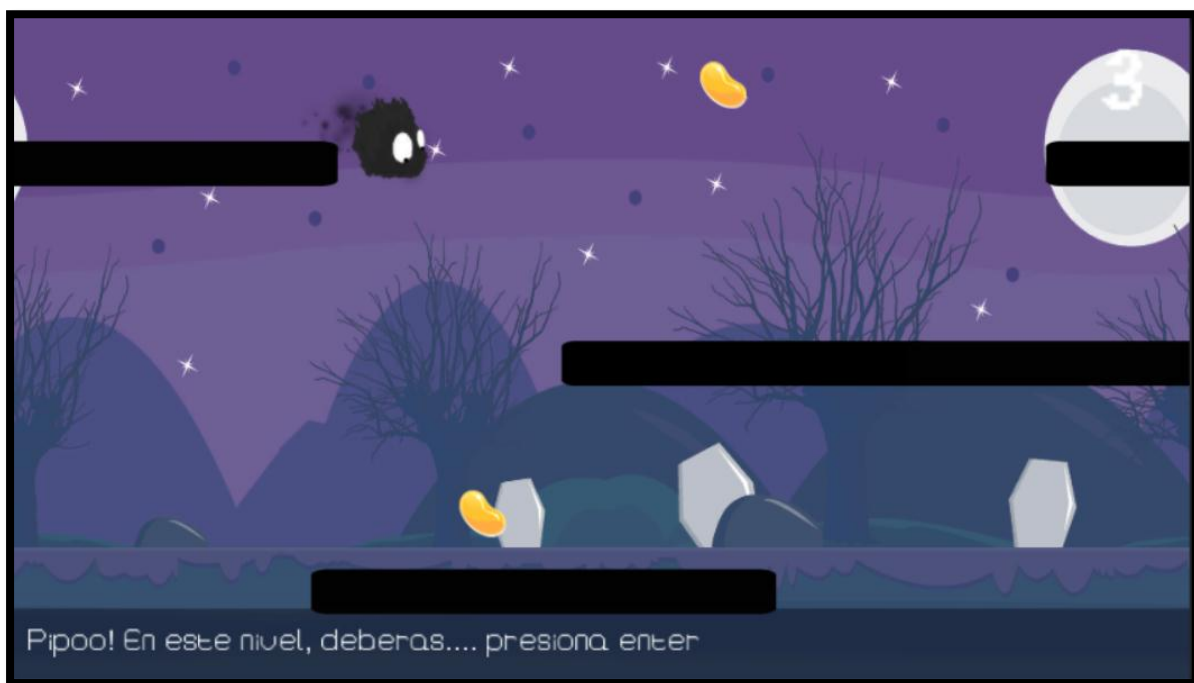
Fuente Autor



Fuente Autor

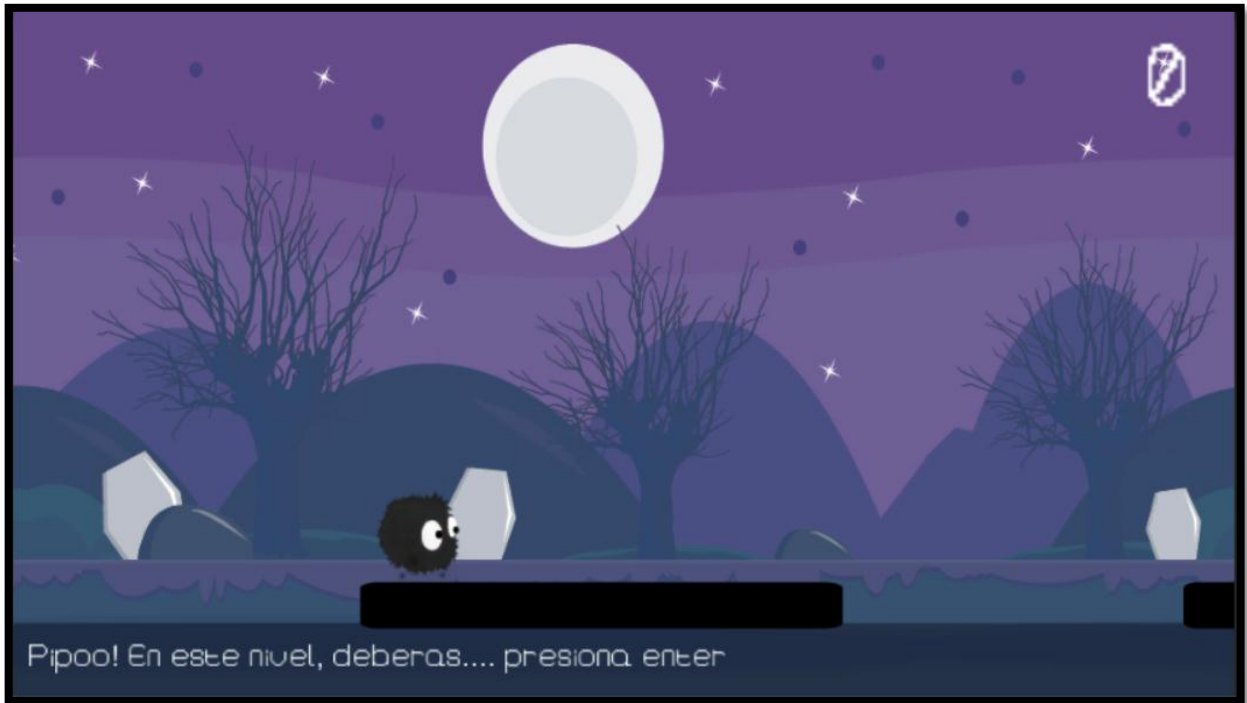
En este nivel, el usuario debe buscar los objetos que le indican en el enunciado, cada objeto cuenta con un valor de 1 punto por obtenerlo. Una vez se atrape el usuario el enunciado indicara el objeto atrapado y la declaración de objetos en programación.

Se atraparan 4 objetos más, luego se pasara al siguiente nivel.



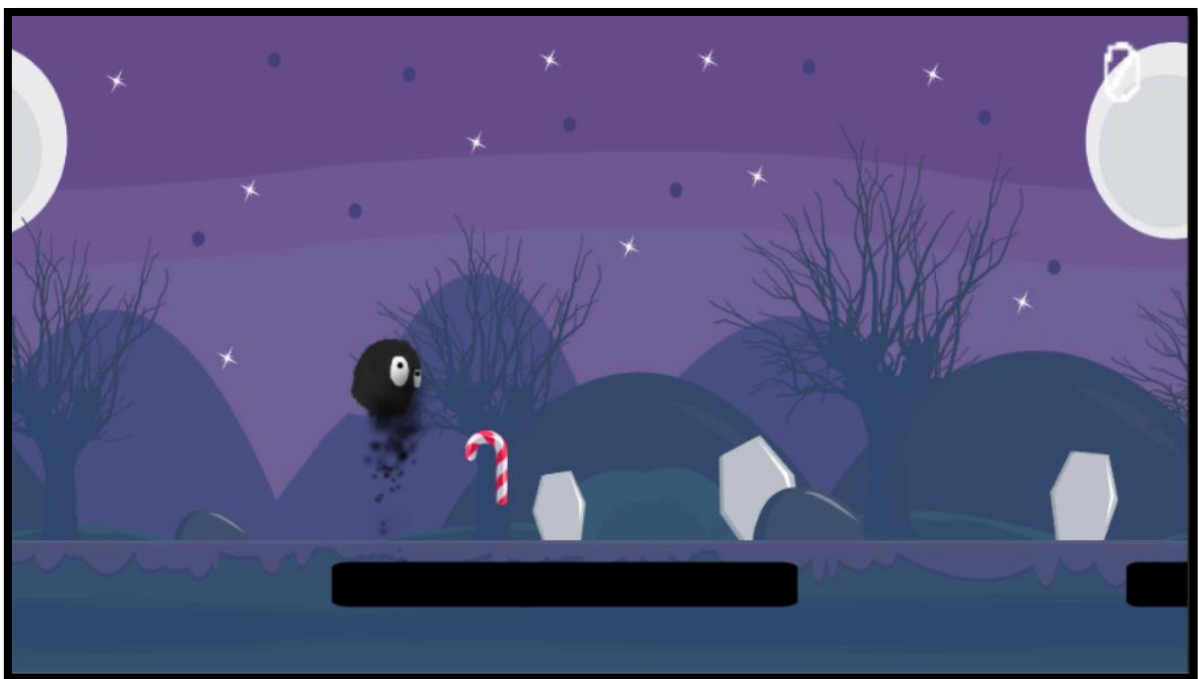
Fuente Autor

El siguiente nivel, se dispondrá del aprendizaje de los ciclos (for, y while), en el cual atraparemos dulces y cada una de ellas tendrán un valor. El primer dulce tendrá un valor de 3, el cual indicara el recorrido de bloques y la explicación del ciclo for por medio de un enunciado.



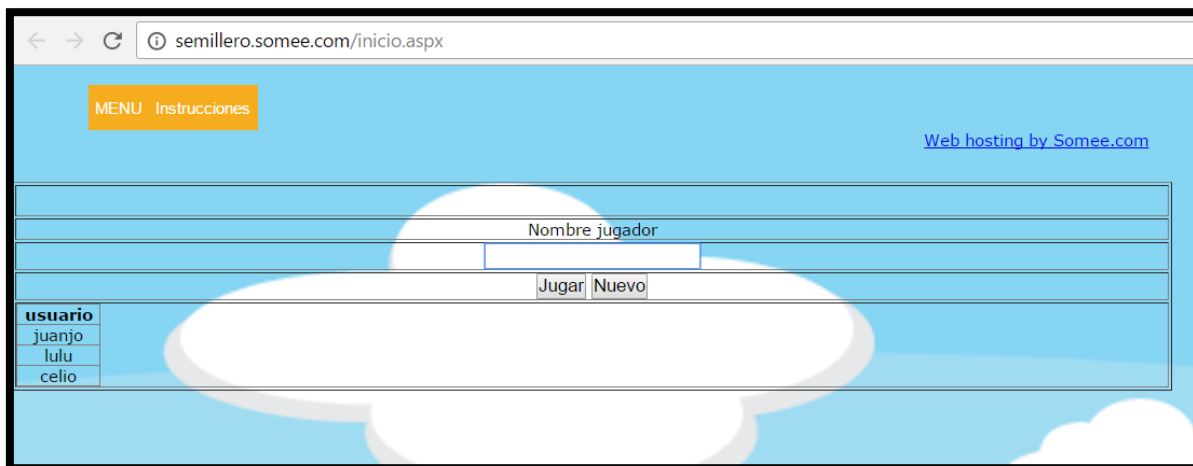
Fuente Autor

El siguiente dulce tendrá un valor de 5, en el cual también se realiza la explicación del ciclo while, mientras se recorre los bloques.



Fuente Autor

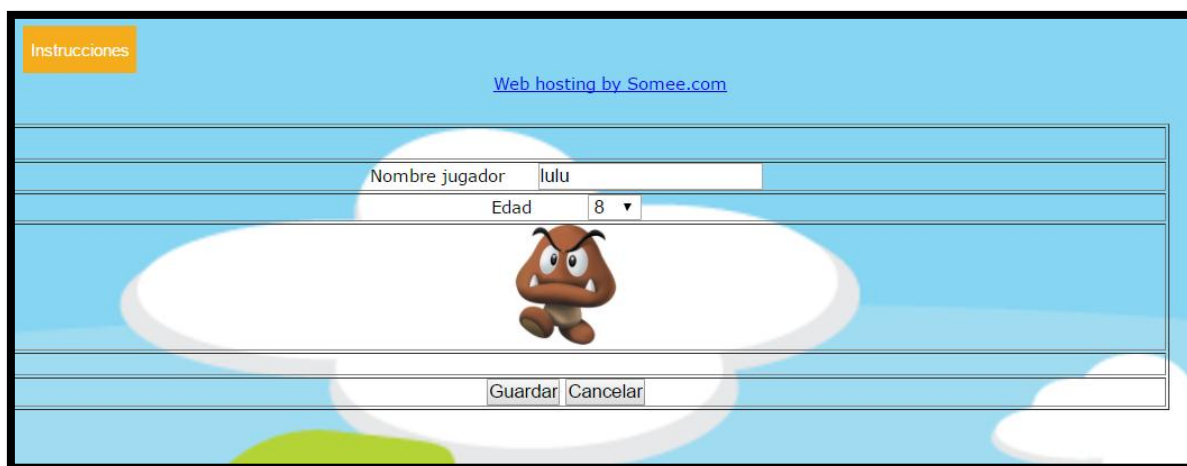
El nivel de las edades de 9 a 12 años, corresponde al recorrido de laberintos por medio de comandos:



usuario
juanjo
lulu
celio

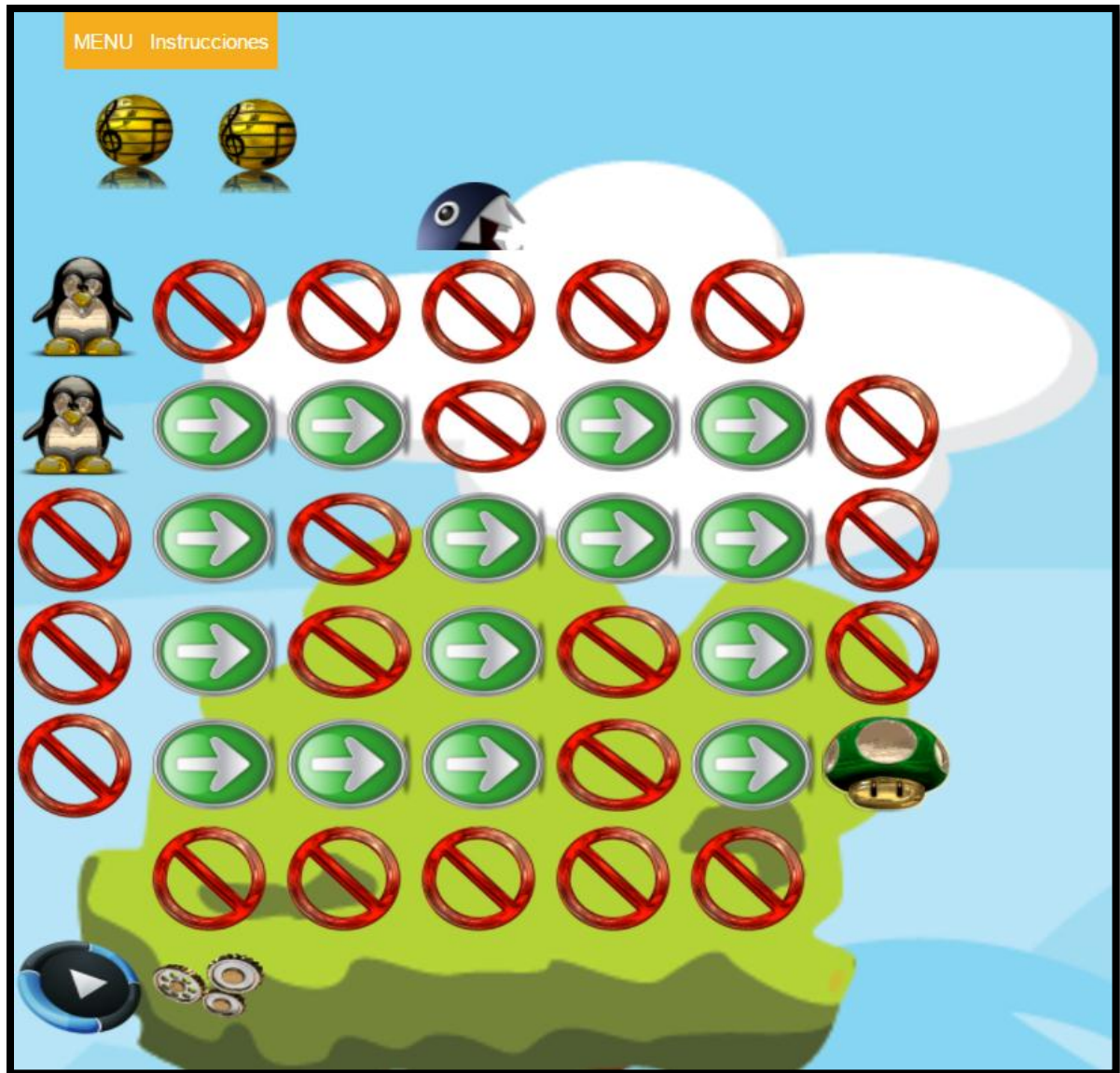
Fuente Autor

Iniciamos el juego, con el logueo de nuestro usuario, en caso de no contar con usuario, en el botón nuevo, podemos generar un nuevo inicio de sesión.



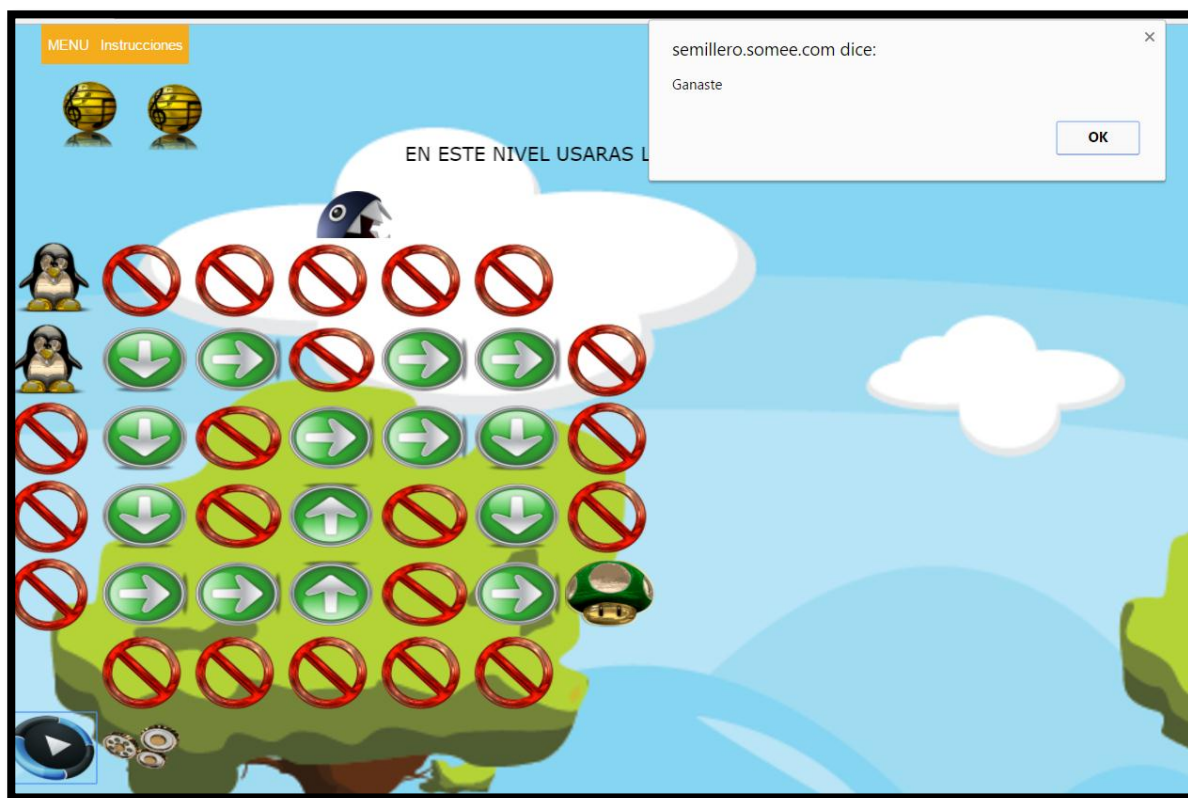
Fuente Autor

Una vez generado el usuario y el ingreso, nos presentara el primer nivel. Se inicia el recorrido del laberinto con un personaje, el cual por medio de secuencias, le indica el camino a completar al personaje.



Fuente Autor

Daremos clic en cada botón verde para indicar su dirección hasta llegar a nuestro destino, que en este caso es el hongo. Luego se determina por medio de un clic en el botón azul para reproducir, el camino que deberá recorrer el personaje.



Fuente Autor

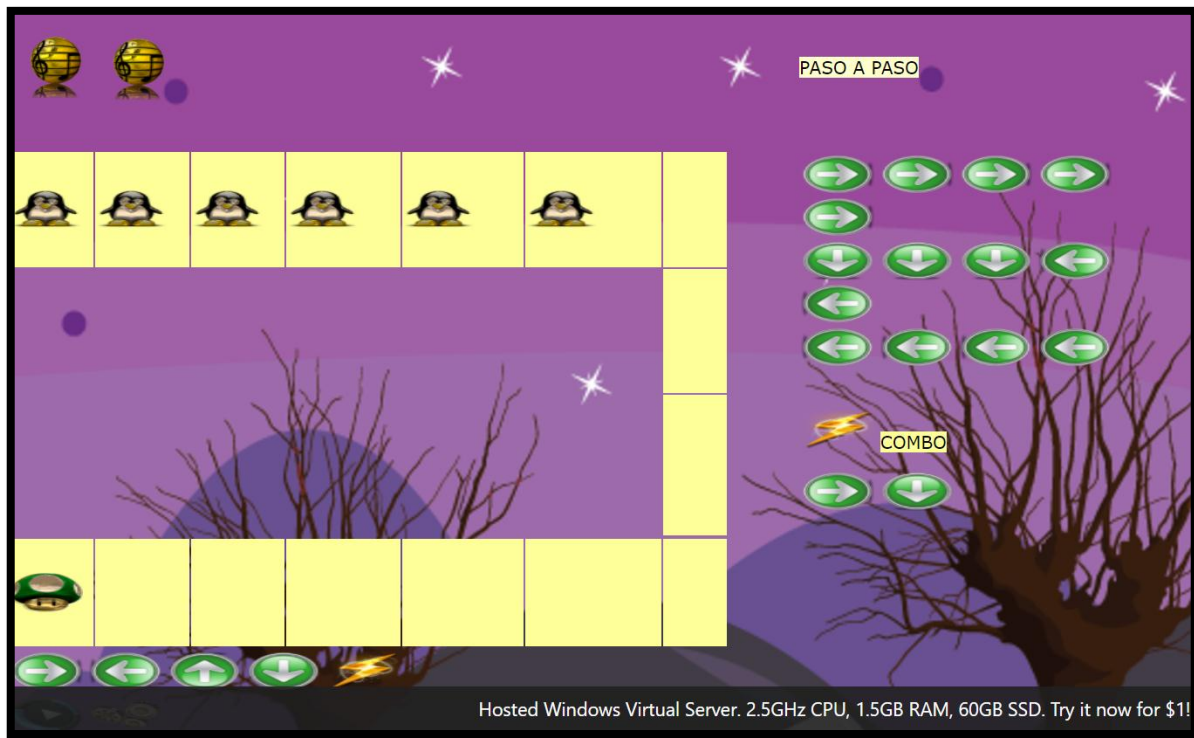
Si el usuario indica el proceso y está mal, el cuadro emergente le hará saber que perdió o que gana en dado caso, con oportunidad de corregir el error.

El nivel 2 del juego, es similar al anterior explicado, sin embargo la diferencia se encuentra en su estructura, ya que es más grande.



Fuente Autor

El ultimo nivel del juego, también consiste en otorgar secuencias al personaje para llegar al destino, incluso usando “combos” combinación de flechas para ejecutar el proceso.



Fuente Autor

De igual forma le indicara al jugador, si perdió o gano el nivel. Le dará play en el botón azul para reproducir y el paso a paso se mostrara en la parte superior derecha.

4. ANÁLISIS DE RESULTADOS Y CONCLUSIONES

4.1. Detalle de resultado

Se evidenciará los resultados de cada etapa que tuvo lugar el software para niños. Se determinaron distintos factores como la interfaz del usuario, la implementación del mismo, su ejecución y lógica de uso. De igual forma niños entre 5 a 12 años realizaron pruebas en estos juegos y se obtuvo resultados positivos sobre el desempeño del software.

- El software cuenta con imágenes llamativas para los niños, de igual forma le acompañan colores, texturas y sonidos.
- El software es amigable con el usuario y posee una interfaz gráfica fácil de comprender y manipular.
- La aplicación cuenta con personajes que le aportan sentido al juego, un mecanismo de historia y es didáctico para el usuario final.
- La ejecución del software se evidencia vía web, lo cual lo hace más asequible para el usuario. De igual forma posee animaciones que le aportan dinamismo al software.
- El software cuenta con lógica básica, para que el usuario aprenda así mismo a programar, mediante la secuencias de procesos o eventos.
- El software cuenta internamente mediante código, con lógica robusta, para que dicha aplicación realice las funciones y requisitos que debe seguir según los eventos que escoja el usuario.

Codificación de programas

Programa	Descripción	Módulo de Proceso Afectado	Tipo Usuario
PlayerController_m.cs	Este programa le permite al personaje realizar acciones dinámicas.	Modificación y Validación	Usuario
ActivateELine.cs	Este programa permite retener las líneas de los textos que se importan.	Modificación y Validación	Sistema
ActivateManager.cs	Este script es el que permite visualizar los textos importados.	Modificación y Validación	Sistema
BotonJugar.cs	Este script permite pasar al siguiente nivel desde cualquier escena.	Validación	Usuario
Puntuacion.cs	Este script permite actualizar la puntuación al	Validación	Sistema

	momento de atrapar un objeto o fruta		
ItemObject.cs	Este script permite asignar un valor según el objeto y enviarlo como notificación al script de puntuación, para su actualización.	Validación y Modificación	Sistema
SeguiPersonaje.cs	Este script le permite al a mainCamara seguir el movimiento del personaje.	Validación	usuario
Generador.cs	Este script permite generar aleatoriamente bloques y frutas.	Validación	Sistema
Destructor.cs	Este script permite la desctrucion de los objetos que quedan fuera de la maincamara.	Modificación y Validación	Sistema

ActivarGameOver.cs	Permite la habilitación de la cámara de “Perdiste”	Validación	Usuario
Bloque.cs y Item_new.cs	Estos scripts permite la creación de los prefabs de bloques y frutas aleatoriamente.	Modificación y Validación	Sistema
ScrollFondo.cs	Permite la animación de movimiento en el segundo nivel y en el inicio del juego.	Validación	Usuario
BoxManager.cs	Permite guardar en un panel los textos que se mostraran en el juego.	Validación	usuario
inicio.asp	Este programa permite la interacción inicial del usuario con la interfaz, así mismo, permite	Validación	usuario

	el redireccionamiento hacia la opción que el usuario elija.		
juegos1.asp	Este programa permite mostrar al usuario la interfaz gráfica del software en este primer nivel.	Validación	Usuario
juegos1.cs	Este programa permite la modificación de los parámetros del juego.	Validación y Modificación	usuario
juegos2.asp	Este programa permite mostrar al usuario la interfaz gráfica del software en este primer nivel.	Validación	Usuario
juegos2.cs	Este programa permite la modificación de los parámetros	Validación y Modificación	usuario

	del juego.		
juegos3.asp	Este programa permite mostrar al usuario la interfaz gráfica del software en este primer nivel.	Validación	Usuario
juegos3.cs	Este programa permite la modificación de los parámetros del juego.	Validación y Modificación	usuario
conectar.cs	Esta clase permite reutilizar código para la asignación del a propiedad imageurl para los distintos imagebutton de los niveles del juego.	Validación	sistema

4.1.1.Pruebas de Función

A continuación se verificaran las pruebas de función que cuenta el aplicativo, se verificaran los casos en el que el usuario pierde o gana, o escoge equivocadamente un objeto.

NOMBRE PROYECTO	PIPOO 1.0
FECHA REPLICA ERROR	N/A
FEHCA ELABORACION DE LA PRUEBA	18/04/2017
MODULO QUE AFECTA:	Escena 1 y 2 Juego (5 a 8 años)
INGENIERO RESPONSABLE	Luisa Fernanda Martinez – Orlando Molina

- Escena 1 (primer nivel del juego): en este módulo, se tiene en cuenta los objetos que el usuario debe atrapar, según su color. Sin embargo si este se equivoca, el sistema mostrara un mensaje de “Objeto equivocado”.

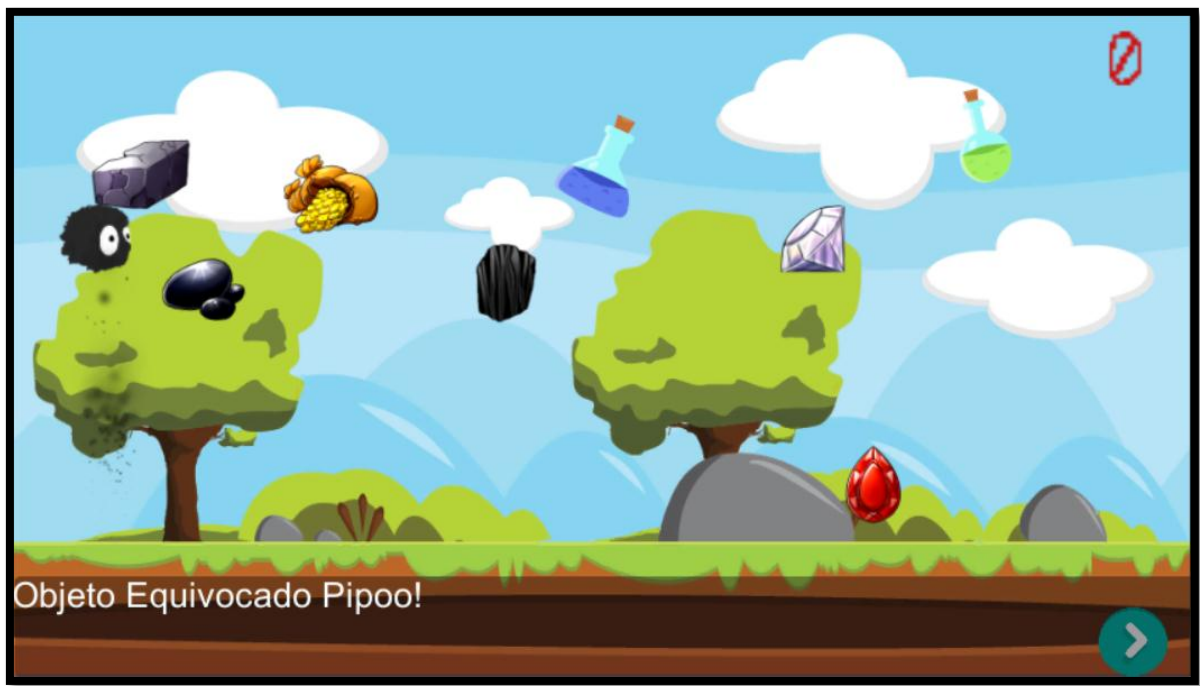


Imagen 6 Fuente Autor

- Escena 1 (primer nivel del juego): De igual forma, una vez atrapados los ítems, el juego deberá indicarle al usuario que ha ganado.



Imagen 7 Fuente Autor

- Escena 2 (Segundo nivel del juego): En este módulo, hay una serie de textos en la parte inferior del juego. Los cuales se van mostrando mediante el usuario avanza en el nivel.

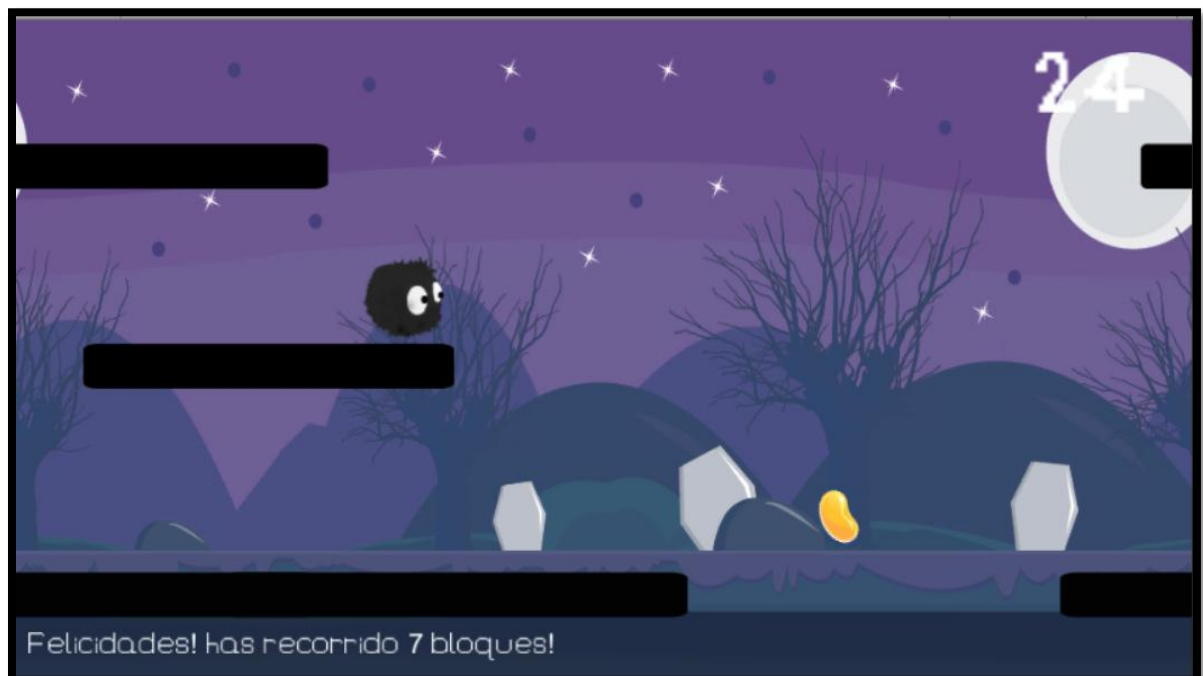


Imagen 8 Fuente Autor

- Escena 2 (Segundo nivel del juego): De igual forma el juego le indicara que ha perdido más el puntaje obtenido.



Imagen 9 Fuente Autor

NOMBRE PROYECTO	PIPOO 1.0
FECHA REPLICA ERROR	N/A
FEHCA ELABORACION DE LA PRUEBA	18/04/2017
MODULO QUE AFECTA:	Nivel 1, 2 y 3 (8 a 12 años)
INGENIERO RESPONSABLE	Luisa Fernanda Martinez – Orlando Molina

- En el inicio de este nivel, solicitara el ingreso de un usuario, si este no existe el juego lo mostrara en una ventana emergente, por lo tanto, el usuario deberá dar clic sobre el boto nuevo y registrar el usuario, para su ingreso.

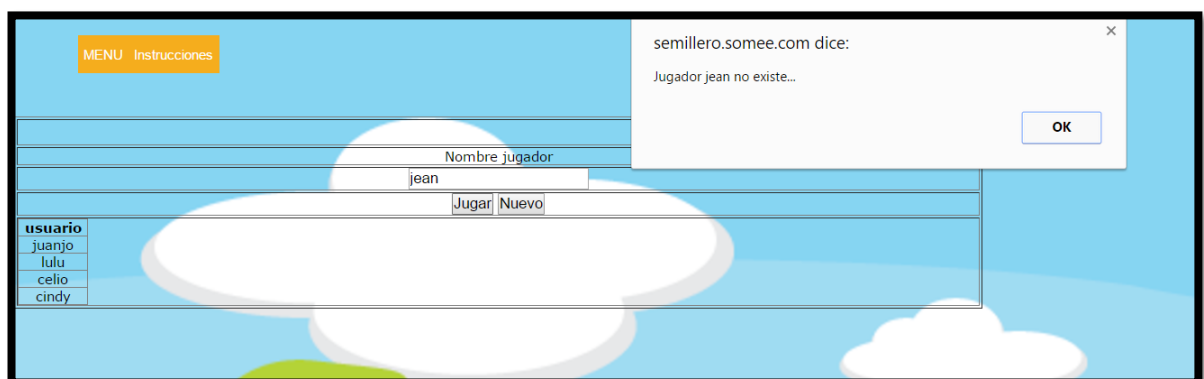


Imagen 10 Fuente Autor

- En el primer nivel, al seguir las indicación del juego, si el usuario pierde, deberá aparecer un cuadro emergente que le indique perdió y le muestre el camino recorrido por el usuario.



Imagen 11 Fuente Autor

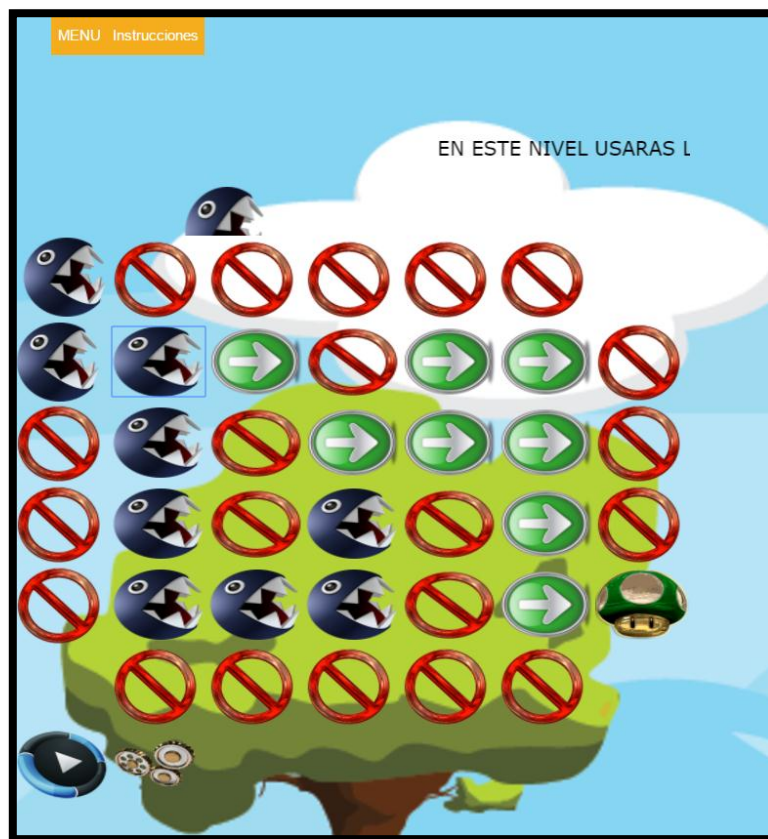


Imagen 12 Fuente Autor

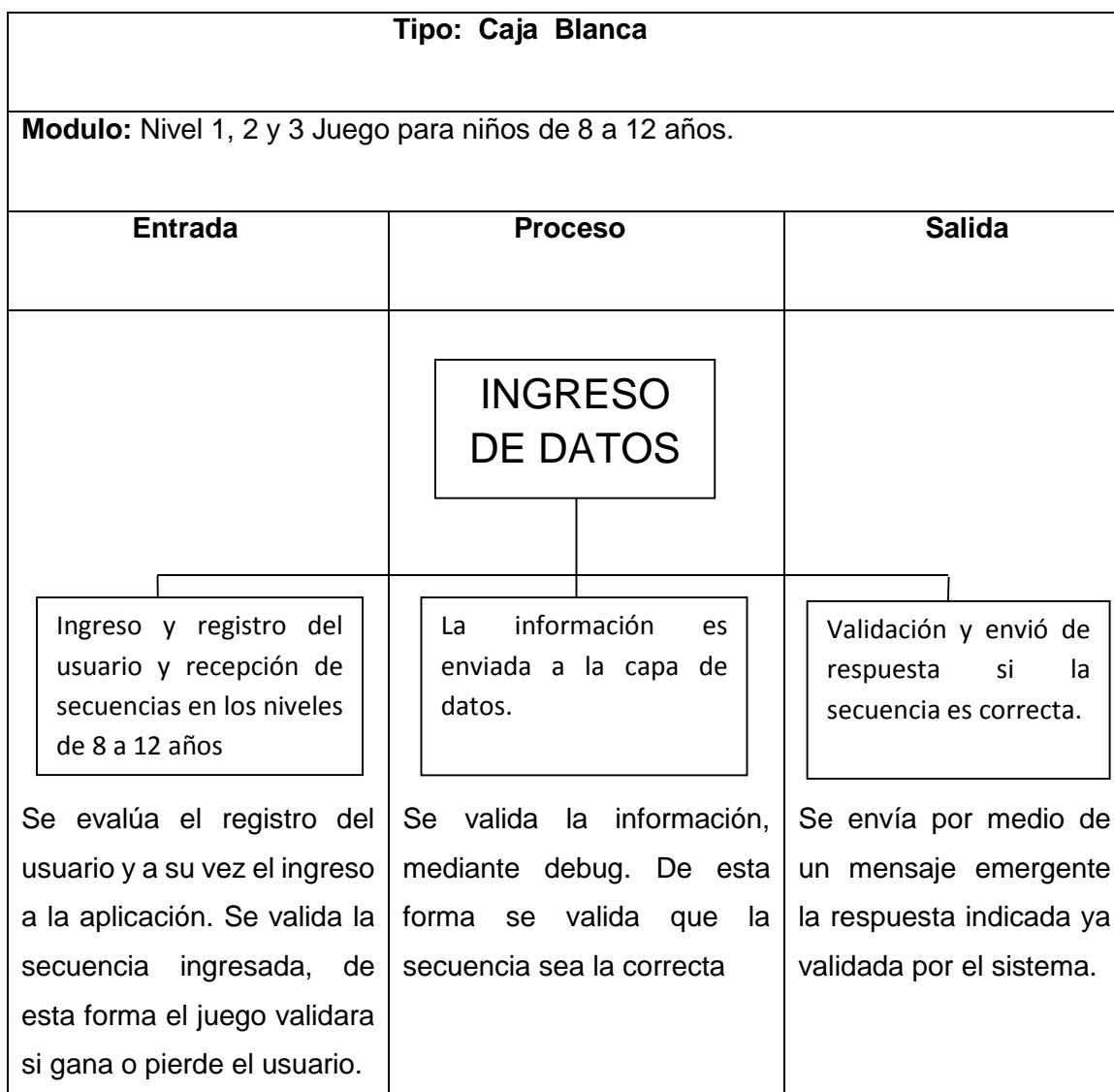
- De igual forma si el usuario gana, el juego le indicara que ha ganado y le mostrara el camino recorrido por el usuario.



Imagen 13 Fuente Autor

4.1.2.Pruebas de Caja Blanca

Mediante la verificación de las pruebas de caja blanca, se evidencia la trazabilidad que el juego tiene entre las escenas y niveles, de esta forma se evalúa la fluidez de la información.



4.1.3.Pruebas Modulares

Se tuvo en cuenta la integración de cada nivel. La validación de la información presentada por el sistema, brindándonos información clara y especifica de secuencias y procedimientos.

4.1.4. Pruebas del Sistema.

Estas pruebas se validaron verificando el rendimiento del juego. Este juego está desarrollado en dos entornos de programación, por lo que se valida que cada uno independientemente funcione correctamente. De igual forma al integrarlos en un sistema web que unifica los niveles por edades, se evidencia que ambos funcionen correctamente. Se valida sonido, movimientos del personaje y la recepción de secuencias.

Pruebas de Rendimiento. Se valida el funcionamiento del Juego en la web, sin verse afectada ninguna función o proceso.

Pruebas de Consistencia. Se validó el funcionamiento de la aplicación, mediante el uso del juego por parte de varios usuarios, de esta forma se confirmó el correcto y continuo funcionamiento juego.

4.1.5. Prueba de Interfaz.

En estas pruebas se valida los personajes y el sonido de las escenas. Se verifica que en cada nivel el personaje siempre se muestre y continúe con el proceso de visibilidad aun cuando se muestra el camino recorrido en el juego de niños de 8 a 12 años. De igual forma en el juego de los niños de 5 a 8 años, se evidencia el correcto funcionamiento de las animaciones, sonidos y movimiento del personaje en ambos niveles.

4.1.6. Prueba de Calidad.

Se realiza la verificación del módulo de ingreso para recibir la respuesta de negación por parte del juego indicando que el usuario no existe. De igual forma se valida en los niveles de los niños de 5 a 8 años el textview donde muestra el puntaje. Se realiza cambios en la asignación de la variable de “puntaje” y se ejecuta el juego para validar que este si este sumando el valor indicado por cada objeto. Se valide que el conteo sea positivo aun cuando se cojan los objetos en diferente orden.

4.1.7. Tipos de pruebas realizadas

Tabla 8 Tipos de pruebas realizadas

Tipo de pruebas generales	SI Cumple	NO Cumple
Acceso al sistema de acuerdo al perfil y a los parámetros definidos.	X	
Acceso a cada uno de los Módulos que conforman el sistema.	X	
Validación de la información por parte del sistema	X	
Ejecución de cada una de las acciones del sistema.	X	
Navegabilidad dentro del sistema	X	
Acceso a los niveles de ayudas	X	
Pruebas de integración	X	
Pruebas de resistencia	X	
Pruebas de rendimiento	X	
Pruebas de compatibilidad	X	
Pruebas de Usabilidad	X	
Pruebas de estilos y diseño	X	

4.2. Análisis de Resultados

Una vez realizadas las validaciones y cada una de las pruebas, se identifica que el juego satisface con las acciones verificadas anteriormente, brindando confiabilidad y funcionalidad en cada uno de los niveles del juego. Se evidencia el correcto funcionamiento de objeto, personaje y contexto en el que se realizó el juego, dando un visto bueno en cada una de las pruebas realizadas y confirmando que el juego cumple con cada una de las especificaciones vistas en las pruebas anteriores en cada nivel del juego.

4.3. Conclusiones

Se concluye que mediante el desarrollo del juego para niños con edades entre 5 a 12 años, se pudo evidenciar que los infantes son muy dados al dinamismo del aprendizaje y esto se logró a través de la creación del juego. También, se evidencio la importancia del aprendizaje con base a una metodología diferente a lo tradicional con aplicación para el mundo real. De igual forma se puede concluir, que se logró lo siguiente:

- Se logró desarrollar un video juego en el cual incentiva a los niños a aprender sobre la programación.
- El nivel de calidad se alcanzó gracias a la realización de las diferentes pruebas, lo que permitió validar el rendimiento y la funcionalidad.
- El diseño del video juego se realizó de forma sencilla, lo cual permitió adaptarse a las edades de los niños (5 años a 12 años).
- Se logró implementar la metodología de Álvaro Galvis Panqueva (ISE), para el desarrollo del juego empleando el diseño y sonido como una de las principales características.
- Se alcanzó un mayor interés por la tecnología y la programación por parte de los niños, evidenciado en las pruebas realizadas a la población de 20 alumnos de un jardín.
- Se logró clasificar los escenarios según edades (5 años a 12 años)

4.4. Recomendaciones

Para futuras versiones se recomienda el desarrollo de niveles con mayor grado de dificultad y una mayor interacción por parte de los niños; lo cual permitirá que este sea mucho más atractivo para el usuario final.

5. Bibliografía

AMBIENTES VIRTUALES DE APRENDIZAJE: UNA METODOLOGÍA PARA SU CREACIÓN Patricia MENDOZA B. Alvaro GALVIS P. Obtenido de http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-106223_archivo.pdf

AMBIENTES EDUCATIVOS PARA LA ERA DE LA INFORMATICA Alvaro H GALVIS PANQUEVA. Obtenido de http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-88541_archivo.pdf

An Educational Software that teaches students computer programming in a 3D environment Alice. Obtenido de <http://www.alice.org/index.php?page=d>

KIDS RUBY Obtenido de www.kidsruby.com/

CODE MONKEY Obtenido de <https://www.playcodemonkey.com/challenges/8>

XATACA Cómo iniciar a un niño en la programación desde cero Obtenido de <https://www.xataka.com/otros/como-iniciar-a-un-nino-en-la-programacion-desde-cero>

Definición de teoría del aprendizaje Julián Pérez Porto. Publicado: 2014. Obtenido de <http://definicion.de/teoria-del-aprendizaje/>

Enumeration Types do not Enumerate! Working around .NET and Language Limitations Sergey Alexandrovich Kryukov 2017. Obtenido de <https://www.codeproject.com/Articles/129830/Enumeration-Types-do-not-Enumerate-Working-around>

Diagramas de componentes de UML: Referencia. Obtenido de <https://msdn.microsoft.com/es-es/library/dd409390.aspx>

UML Diagramas de Paquetes (UML ilustrado) Universidad de los Andes Demián Gutierrez. Obtenido de http://www.codecompiling.net/files/slides/UML_clase_05_UML_paquetes.pdf

Diagramas de Despliegue. Obtenido de <http://www.diadspg.blogspot.com/>

Diagramas de Despliegues. Obtenido de <http://ingenieriasoftwaredos.wikispaces.com/Diagramas+de+Artefactos+y+despliegue?responseToken=2a317998756967c0eb041d05269a0ca4>

Música y Sonidos - Tutorial Unity 2D en español Hagamos Videojuegos

<https://www.youtube.com/watch?v=WZemrM46AGU&t=1420s>

Desarrollo de Videojuegos en Unity: Una Introducción by Universidad de los Andes.

Obtenido de <https://www.coursera.org/learn/desarrollo-videojuegos-unity/home/welcome>

EPSITEC Obtenido de <http://www.ceebot.com/ceebot/language-e.php>

CODE HUNT Obtenido de <https://www.codehunt.com/>

APRENDE A PROGRAMA JUGANDO, MDZ. Obtenido de <http://www.mdzol.com/nota/502024-aprende-a-programar-jugando>

PORTAL DE EDUCACION INFANTIL, Obtenido de <http://www.educapeques.com/estimulapeques/juegos-de-asociaciones-para-mejorar-la-logica-y-memoria.html>

Juegos de Memoria, Lógica y Habilidad para niños. Obtenido de <http://educapeques.com/los-juegos-educativos/juegos-de-memoria-logica-habilidad-para-ninos/portal.php>

CODIGO FLAPY. Obtenido de <https://studio.code.org/flappy/1>

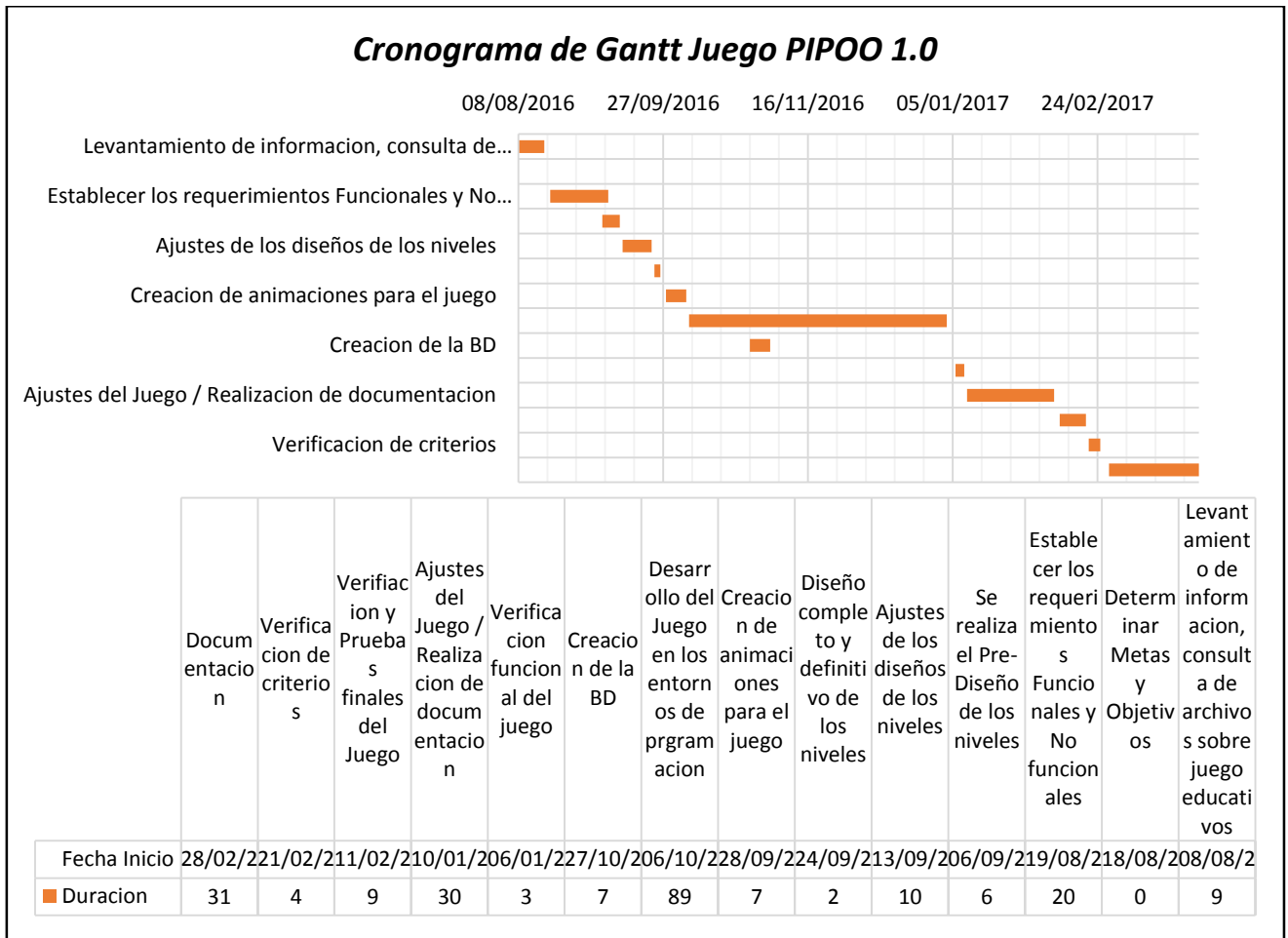
CODE COMBAT. Obtenido de <https://codecombat.com/play>

La Formación Docente para la Innovación Educativa, el caso del currículum de High Scope. Por Roberto Barocio Quijano, 2002.

Visual C# 2012 HOW TO PROGRAM, Quinta Edición. Por Paul Deitel Harvey Deitel, 2013.

ANEXOS

Cronograma del proyecto:



Fuente: Autor